Springer Optimization and Its Applications 92

Fuad Aleskerov
Boris Goldengorin
Panos M. Pardalos *Editors*

Clusters, Orders, and Trees: Methods and Applications

In Honor of Boris Mirkin's 70th Birthday

Springer

Springer Optimization and Its Applications

VOLUME 92

Aims and Scope

Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics, and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository work that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.

For further volumes:
http://www.springer.com/series/7393

Fuad Aleskerov • Boris Goldengorin
Panos M. Pardalos
Editors

Clusters, Orders, and Trees: Methods and Applications

In Honor of Boris Mirkin's 70th Birthday

Springer

Editors
Fuad Aleskerov
Higher School of Economics
National Research University
Moscow, Russia

Boris Goldengorin
Department of Operations
University of Groningen
Groningen, The Netherlands

Panos M. Pardalos
Department of Industrial
and Systems Engineering
University of Florida
Gainesville, FL, USA

ISSN 1931-6828 ISSN 1931-6836 (electronic)
ISBN 978-1-4939-0741-0 ISBN 978-1-4939-0742-7 (eBook)
DOI 10.1007/978-1-4939-0742-7
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2014939298

Mathematics Subject Classification (2010): 68Pxx, 68Qxx, 68Rxx, 68Txx, 68Uxx, 90-XX

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book is a collection of papers written for the International Workshop *Clusters, orders, trees: Methods and applications* on the occasion of the 70th Anniversary of Boris Mirkin, Professor at National Research University Higher School of Economics (NRU HSE), Moscow Russia, and Birkbeck College University of London, UK, which was held in NRU HSE Moscow, 12 and 13 December 2012, with the following schedule.

12 December 2012

11.00–11.20 Opening

11.20–12.00 Fuad Aleskerov (NRU HSE, Moscow, Russia). Interval orders, semiorders, and their numerical representation.

12.00–12.30 Katya Chernyak (NRU HSE, Moscow, Russia). Scoring extent of similarity between a string and text using suffix tress: method and applications.

12.30–13.00 Yulia Veselova (NRU HSE, Moscow, Russia).The manipulability index in the IANC model.

13.00–15.00 Break

15.00–15.40 Fred Roberts (Rutgers University, NJ, USA). Meaningless statements in landscape ecology and sustainable environments.

15.40–16.20 Vladimir Makarenkov (Université de Québec, Montréal, Canada). Building trees encompassing horizontal transfer events: applications in evolution and linguistics.

16.20–16.50 Break

16.50–17.30 Trevor Fenner (University of London, UK). Lifting algorithm in a tree and its applications.

17.30–18.00 Andrey Shestakov (NRU HSE, Moscow, Russia). Least squares consensus clustering.

13 December

11.00–11.40 Valery Kalyagin (NRU HSE, Nizhniy Novgorod, Russia). Mining market data: Russian stock market.

11.40–12.20 Sergei Archangelski, Ilya Muchnik (Rutgers University, NJ, USA). Clustering in registration of 3D point clouds.

12.20–13.00 Ito Wasito (University of Indonesia). Least-squares imputation of missing data entries.

13.00–14.30 Break

14.30–15.10 Panos Pardalos (University of Florida, USA). High dimensional data classification.

15.10–15.50 Fred McMorris (IIT, Chicago, USA). The majority decision function on median semilattices.

15.50–16.10 Break

16.10–16.50 Susana Nascimento (Universidade Nova de Lisboa, Portugal). Data recovery fuzzy clustering: proportional membership and additive spectral methods.

16.50–17.30 Boris Goldengorin (NRU HSE, Moscow, Russia). Mixed tools for market analysis and their applications

This workshop has reflected some past and present Boris Mirkin's research activities. Boris Mirkin has contributed to the development of all three concepts with outstanding contributions.

In clustering, among others are

1. deriving and using distance between partitions (1969, in Russian) predating the popular Rand index (1971) which is the complement of Mirkin's distance to unity,
2. proposing the so-called qualitative fuzzy analysis model and methods (1976, in Russian) which includes the popular additive clustering model by Shepard and Arabie (1979) and an effective one-by-one approach for identifying it, and
3. principal cluster analysis model and method (1987, in Russian) later converted by him into the Separate-and-Conquer strategy (1998) and then to Anomalous Pattern clustering and Intelligent K-Means methods (2005).

A joint paper by R. Amorim and B. Mirkin in this volume relates to a further extension of the latter approach to Minkowski distances and weighted variables. A joint paper by Mirkin and Shestakov shows how effective can be an approach to consensus clustering outlined by B. Mirkin and I. Muchnik back in 1981 (in Russian), in comparison to recent approaches.

In ordering, among others are:

1. a characterization of the interval orders as those irreflexive relations whose images are linearly ordered over settheoretic inclusion (1969, in Russian), and
2. extension of the Arrow consensus between rankings approach to equivalence relations (1971) and then any relations (1973, in Russian, 1979 in English), and using Arrow axioms for characterization of what he, and then the others, calls federation consensus rules (1979, in Russian).

In trees, among others are:

1. a biologically meaningful mapping of gene phylogenetic trees onto a species evolutionary tree (Mirkin–Muchnik–Smith model 1995) and
2. split vector bases corresponding to binary hierarchies (1996).

A paper by K. Chernyak and B. Mirkin describes further developments in text analysis based on the usage of two tree-related concepts, suffix tree, and taxonomy. It should be pointed out that the concepts of cluster, order, and tree are in the core of all efforts in data analysis, and these are to remain in the core for the foreseeable future, because they are in the core of any structuring attempts.

Of course, no real-world application can be developed without measuring features and relations, and B. Mirkin has contributed to this, as well.

Examples include:

1. matrix-based correlation and association measures in the space of mixed scale variables (1982, in Russian) and
2. least-squares methodology for imputation of missing values. Several papers in this volume can be attributed to this direction.

Accordingly, we divided all the contributions in three sections:

(a) classification and cluster,
(b) order and tree, and
(c) measurement.

In addition to Boris Mirkin's startling Ph.D. results in mathematical logic and algebra, Mirkin's groundbreaking contributions in various fields of decision-making theory and practice have marked the fourth quarter of the twentieth century and beyond. Boris has done pioneering work in group choice, mathematical psychology, clustering, data mining and knowledge discovery which are activities oriented towards finding nontrivial or hidden patterns in data collected in databases. Boris Mirkin has published several books, such as: The Group Choice Problem (in Russian, 1974), Analysis of Quality Indicators (in Russian, 1976), Graphs and Genes (in Russian, co-authored with S.N. Rodin, 1977), Group Choice (Wiley-Interscience, 1979), Analysis of Quality Indicators and Structures (in Russian, 1976), Graphs and Genes (Springer, co-authored with S.N. Rodin, 1984), Clusters in Social- Economics Research (in Russian, 1985), Mathematical Classification and Clustering (Kluwer, 1996), Clustering for Data Mining: A Data Recovery Approach (Chapman and Hall/CRC, 2005; Second Edition, 2012), Core Concepts in Data Analysis: Summarization, Correlation and Visualization (Undergraduate Topics in Computer Science) (Springer, 2011).

Our special thanks to all reviewers who made a crucial contribution to the scheduled production of this volume. Here we would like to list all of them: Fyad Aleskerov, Rozenn Dahyot, Anuška Ferligoj, Boris Goldengorin, Dmitry Ignatov, Friedrich Leisch, Vladimir Makarenkov, Boris Mirkin, Sergei Obiedkov, Panos M. Pardalos, Niel J le Roux, and Yulia Veselova.

This volume contains the collection of papers reflecting many developments in theory and applications rooted by Boris' fundamental contribution to the state of the art in group choice, mathematical psychology, clustering, data mining, and knowledge discovery. Researches, students, and engineers will benefit from new knowledge discovery techniques.

Moscow, Russia — Fuad Aleskerov
Groningen, The Netherlands — Boris Goldengorin
Gainesville, FL, USA — Panos M. Pardalos

Contents

Three and One Questions to Dr. B. Mirkin About Complexity Statistics .. 1
Igor Mandel

Part I Classification and Cluster

A Polynomial Algorithm for a Class of 0–1 Fractional Programming Problems Involving Composite Functions, with an Application to Additive Clustering 13
Pierre Hansen and Christophe Meyer

Experiments with a Non-convex Variance-Based Clustering Criterion 51
Rodrigo F. Toso, Evgeny V. Bauman, Casimir A. Kulikowski, and Ilya B. Muchnik

Strategy-Proof Location Functions on Finite Graphs 63
F.R. McMorris, Henry Martyn Mulder, and Fred S. Roberts

A Pseudo-Boolean Approach to the Market Graph Analysis by Means of the p-Median Model 77
Boris Goldengorin, Anton Kocheturov, and Panos M. Pardalos

Clustering as an Approach to 3D Reconstruction Problem 91
Sergey Arkhangelskiy and Ilya Muchnik

Selecting the Minkowski Exponent for Intelligent K-Means with Feature Weighting 103
Renato Cordeiro de Amorim and Boris Mirkin

High-Dimensional Data Classification 119
Vijay Pappu and Panos M. Pardalos

Algorithm FRiS-TDR for Generalized Classification of the Labeled, Semi-labeled and Unlabeled Datasets 151
I.A. Borisova and N.G. Zagoruiko

From Separating to Proximal Plane Classifiers: A Review 167
Maria Brigida Ferraro and Mario Rosario Guarracino

A Note on the Effectiveness of the Least Squares Consensus Clustering .. 181
Boris Mirkin and Andrey Shestakov

Part II Order and Tree

Single or Multiple Consensus for Linear Orders 189
Alain Guénoche

Choice Functions on Tree Quasi-Orders 201
F.R. McMorris and R.C. Powers

Weak Hierarchies: A Central Clustering Structure 211
Patrice Bertrand and Jean Diatta

Some Observations on Oligarchies, Internal Direct Sums, and Lattice Congruences 231
Melvin F. Janowitz

Thinking Ultrametrically, Thinking p-Adically 249
Fionn Murtagh

A New Algorithm for Inferring Hybridization Events Based on the Detection of Horizontal Gene Transfers 273
Vladimir Makarenkov, Alix Boc, and Pierre Legendre

Part III Measurement

Meaningful and Meaningless Statements in Landscape Ecology and Environmental Sustainability 297
Fred S. Roberts

Nearest Neighbour in Least Squares Data Imputation Algorithms for Marketing Data 313
Ito Wasito

AST Method for Scoring String-to-text Similarity 331
Ekaterina Chernyak and Boris Mirkin

Improving Web Search Relevance with Learning Structure of Domain Concepts 341
Boris A. Galitsky and Boris Kovalerchuk

Linear Regression via Elastic Net: Non-enumerative Leave-One-Out Verification of Feature Selection 377
Elena Chernousova, Nikolay Razin, Olga Krasotkina, Vadim Mottl, and David Windridge

The Manipulability Index in the IANC Model 391
Yuliya A. Veselova

Three and One Questions to Dr. B. Mirkin About Complexity Statistics

Igor Mandel

Abstract I share my personal thoughts about Boris Mirkin and, as a witness of his long-term development in data analysis (especially in the area of classification), pose several questions about the future in this area. They are: about mutual treatment of the variables, variation of which has very different practical importance; relationship between internal classification criteria and external goals of data analysis; and dubious role of the distance in clustering in the light of the last results about metrics in high dimensional space. The key question: the perspective of the "complexity statistics," similarly to "complexity economics."

Keywords Data analysis • Classification • Clustering • Distances • Complexity • Sociosystemics

> . . . animals can be divided into (a) those belonging to the Emperor, (b) those that are embalmed, (c) those that are tame, (d) pigs, (e) sirens, (f) imaginary animals, (g) wild dogs, (h) those included in this classification, (i) those that are crazy acting (j), those that are uncountable (k) those painted with the finest brush made of camel hair, (l) miscellaneous, (m) those which have just broken a vase, and (n) those which, from a distance, look like flies. J.L. Borges "The Analytical Language of John Wilkins" (translated by W. Fitzgerald), 1952.

> *All Job Types: Biotech; Construction; Customer Service; Executive; Entry Level –New Grads; Inventory*. Fragment of the list from the leading job searching website Careerbuilder.com, 2013.

The famous Borges' classification of animals is far from being just a sharp parody on a pseudo-science—it is a reality in many situations all over the place. In a small example in a second epigraph, at least four foundations are used to describe jobs—by industry (Biotech, Construction), by specific human activity in any

I. Mandel (✉)
Telmar Group Inc., 711 Third Avenue, New York, NY 10017, USA
e-mail: imandel@telmar.com

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_1,

industry (Customer Service, Executive); by education (Entry Level—New Grads), and by production function in any industry (Inventory). At a first glance, such a classification looks absurd and non-mutually exclusive (while mutual exclusivity is what everybody expects from the very term "classification"). Executive could be anywhere, and what Finance CEO should look for: his peer Executives in any branch, or specific position in Finance? But on the other hand, the website has existed for many years, it is very popular, and no one seems to care about such obvious inconsistencies. It tells a lot about human ability to work in an uncertain situation—the fact confirmed many times in modern psychology. In particular, this feature, together with others, undermines the whole concept of the human rationality (assumed in Homo Economicus from the classical economics). Classification theory assumes that some dividing lines exist and the problem is how to detect them. Reality asserts that these lines are almost always illusionary and/or imaginary.

Boris Mirkin has been thinking about this dilemma for the last 45 years to the admiration of many observers, myself included. For these reasons it is a good time to ask the main character of the book some questions about the activity, which excites him for such a long time—presumably, if he knows the answers, it could be a nice present because it would raise his self-esteem, but if he doesn't—it would be even a better present, because what could be more valuable for the real scholar than new questions? But before I get to the questions, I would make a short introduction of the addressee as I see him.

1 Three Constants in Boris Mirkin's Life

In 1974, after having been discharged from the Army, I was happy to come back to normal life, which meant to me to do something about "pattern recognition," as was a popular way to refer to the field at that time. I intensively and un-systematically read on the subject what I could find in Alma-Ata, the capital of Kazakhstan, wrote letters to some especially interesting to me specialists and, in the end, decided to go to Novosibirsk Academgorodok to visit Dr. B. Rozin who was very active in this area. While walking by the Institute of Economics there with my friend, we noticed an office in which a certain Head of Lab Dr. B. Mirkin was sitting, according to the plaque. This name did click in my memory (I had read his first book already). So we knocked at the door. A joyful and smiling, rather young person stepped up and asked something like, "How can I help you?" (now I caught myself thinking that his look practically hadn't changed in all these years . . .). A smile bordering with a pleasing laugh and readiness to talk to strangers at a workplace were not quite common at that time in Russia (it seems still are not). We smiled and laughed in response. Then in a one hour-long conversation we learned a lot of things about each other without any particular purpose, as it does happen in Russia—certain patterns, maybe, were just recognized. So, this is how it started.

Since that time we never lived in the same geographical area, but somehow often met in Moscow, Alma-Ata, Yalta, Kiev, Paris, Rochester (NY), New York,

Princeton, Highland Park (NJ), Washington DC, London (and Moscow again just in 2012!). Boris wholeheartedly supported my book [1] and wrote a nice introduction to it; we published a review of current clustering approaches in the beginning of 1990s; and we discussed tens of topics in the course of all these years on the phone and Skype and so on.

The best way to learn something about someone's personality is to observe what he or she is doing when circumstances are changed. Most people follow the mainstream, with all its twists. Yet some follow their goals regardless of these fluctuations. As long as I know Boris, he belongs to the minority. He had changed a dozen positions in five to six countries in the turbulent times from the end of the 1980s, but *one* thing remained *constant*—he worked and reflected his work in his books. From the first one, "Group Choice" (1974, in Russian), which elevated him to the top of the analytical community and triggered our meeting, to the latest one "Clustering" (2012)—he always wrote books (I do not know how many—my lower estimate is ten). Of course, he wrote many tens of articles as well, but his passion to write books seems to me unprecedented. I vividly remember how much it cost me to write just one and can imagine what it is to have ten, on different topics, of the very high quality, highly original, and almost all as a single author. One may expect that a person capable of doing that is a kind of gloomy scientist thinking only about writing mandatory number of pages per day and will be way off.

In fact, all our talks started and ended with jokes and laughing, which seems to be the *second constant* element in Boris' life. He has not only permanently produced jokes himself, but vividly reacted to those of others. It was the main reason why most of our scientific discussions quickly went in an unpredictable direction, and ultimately the original topic could disappear entirely (but new ones emerged). As a result, we published only one joint work, while another one is still buried under a pile of jokes for the past 5 years.

The *third*, and the most surprising *constant*, is Boris' tolerance. Since we both live in the so-called interesting times, what the Chinese wish for their enemies, I could expect to hear extreme opinions and complaints, from the left, the right, the top, and the bottom—and I did hear much of them indeed. But never from Boris. His tolerance was not only towards the politics, but in fact towards everything; his belief in a good side of the human nature, I'm sure, is a key in helping him to overcome many troubles and to keep his first and second constants (i.e., writing and laughing) alive. A wonderful painting by Alexander Makhov hangs on the wall in Boris' Moscow apartment—a big fish is spasmodically bent at the beach in the attempt to get off the hook from the fishing line. I definitely saw it as a symbol of tragedy and torture—but Boris suddenly said that he bought it because he sees there the unshakable will to struggle for life. And I agreed that this interpretation was also possible—actually, might be the only one possible.

Knowing the constants of Boris' life, I can ask questions and easily predict Boris' reaction: he would either write a book about the problems, to the pleasure of myself and others; or he would laugh at them, leaving me bewildered; or he would display a tolerant view, following the principle, "let baby play with whatever he wants." So, encouraged by these predictable options, I came up with my questions.

2 Questions About Clustering and More

1. *Classify or not*? Any classification is very tricky, even for one-dimensional data. Where is the borderline between low and middle incomes, between high and very high intelligence, and so on? Where does the idea of "discontinuity" start and the idea of "qualitative difference" end? Can we think that, if data are generated from one source, which has a perfect (say, normal) distribution (an almost impossible situation, at least in my practice), they are homogeneous and need no further division? Many would say, "yes" (the whole logic of kernels approaches in clustering is based on that idea). But let us assume that we have two normal variables of that type—one of the results of measuring of the length of the pen, and another of people's height. In the first case, the standard deviation could be a fraction of a percent of the "real length"; in the second—about 20 cm of the true average height of 170 cm. Why is it intuitively clear that data in the first case are completely homogeneous and describe certain phenomenon with high precision (no one will be interested in the separation of the two tales of the distribution), but in the second case data are heterogeneous, and we do need separate people of 210 cm in height and more from those whose height is less than 120 cm?

   Formally, these two cases are identical, and any density-like procedures would treat them, respectively (either trimming the data to, say, one standard deviation interval or leave as it is, since it is proven that they belong to the same distribution). Statisticians do not relate the standard deviation to the average levels—these, indeed, are two *independent* parameters, and why should we care about their relationship (moreover, when the average is close to zero or negative, the ratio does not exist). But our common sense and real practical needs tell something opposite—variation in relation to the average (something like coefficient of variation) does matter.

   Question one: can we formally and substantively distinguish these situations in clustering, especially remembering that data standardization usually makes all the standard deviations equal? Can we legitimately say that variables, like length in the above example, are to be excluded from analysis?
2. *Classification and the final goal*. On a more fundamental level, any formal classification, like ones produced in cluster analysis, actually have only "intermediate" value, dependent on the type of a problem to be solved, no matter what internal criteria were used in clustering. I found in the literature about 50 different optimization criteria in clustering in 1980s [1]; now, I'm sure, there are many more. But anyway, each classification has just an intermediate value, and the ultimate goal of the process where clustering is just a part having usually no relation to the internal criteria used.

   Let's take, for example, the celebrated K-means procedures, proposed in 1957–1967 by several authors, which were a topic of B. Mirkin's studies in many publications. They are included virtually in all statistical packages and are by far the most popular among clustering algorithms. The idea is to break multidimensional data into K clusters in such a way that the squared sum of

deviations from the cluster centers is minimized. Since it is an NP-hard problem, many heuristic algorithms were proposed and used. Let's say one uses this type of procedure to group the data for so-called market segmentation, which is a typical problem in marketing.

To make it more concrete, a researcher collects data from about 20,000 respondents who answered couple of tens of "attitudinal" questions like: (a) "I try to eat a healthy breakfast every day" (the answer 5 means "completely agree with this statement", 1—completely disagree, and 2–4 are in the middle); (b) "I enjoy being creative in the kitchen"; (c) "I often eat my meals on the run", etc. The aim of a targeting campaign is to sell cereals ZZZ. The purpose of clustering is to break potential customers into five groups where they supposedly have similar attitudes and then target the groups differently, i.e. ultimately apply five different marketing strategies. For example, if cluster 1 has all 5s for questions (a) and (b)—they will get the special offer, stressing the importance of ZZZ for healthy breakfast and enormous number of creative ways to combine ZZZ with everything in a fridge; if in cluster 2 all 5s will be for questions (a) and (c)—the message would be that it is extremely easy to take the healthiest cereals in the world, ZZZ, and eat it on the run, and so on. I deliberately left behind all the questions related to clustering *per se* here (why 5, not 4 clusters; how far the solution is from the global optimum; how beneficial is the fact that we don't need variables normalization, etc.). Let's focus just on one issue: what is the relation between this obtained cluster solution and the final goal of the marketing campaign?

Let us assume, there is a certain (unknown) function linking answers to these questions with probability to buy ZZZ if respondent got the advertising: the probability to buy ZZZ for these with answers 5 for (a) is P(5a); for answers 5 for (b) is P(5b), and so on, like P(4a), P(4b) .... Combination of answers, like (a) = 5 and (b) = 3 also create some kind of unknown probability to buy the product. Ultimately, what marketer needs is to redistribute her limited budget between particular strategies in such a way that probability of buying ZZZ for entire set of potential customers is maximized. They should be then broken into several groups or not broken at all—it may very well be that one strategy may be the best in terms of obtained buying rate per dollar spent. The problem then is: how clustering solution should be formulated in order to meet this external goal.

It is quite clear that K-means minimization has in fact almost nothing to do with it. Probability functions associated with respondents within each cluster can vary wildly regardless of how close objects in cluster are to each other (unless they are all identical). But what kind of relations between the objective function and variables to be used in clustering should exist in order to solve the final, not intermediate problem? In its simplest form, the problem could be presented like that: one has N covariates of X type (independent) and one variable of Y type (dependent). One can estimate the relations between Y and X either on the whole data, or in each cluster (maybe, just via regression). What are the clustering procedures (or strategies) such that the total gain of these two processes (classification and regression estimation) is maximized?

Similar problems have been handled for a long time via piece-wise data approximation procedures and even piece-wise mixed models, where instead of using pre-defined groups, the group membership has to be estimated from the data. I discussed in [1] the so-called *intentional statistical analysis*—a general way to make statistical procedures, including clustering, oriented to the final goal, which is usually expressed in monetary or other non-statistical terms. The main idea was to reorganize data in such a way that this external goal is met. It can easily happen that to reach a goal we don't even need a whole data set (keep aside to cluster all data), as it happened in campaigns targeting only the first percentile of the population ordered by their propensity to respond to the advertisement. It seems this problem did not lose its importance.

For example, the authors of "Goal oriented clustering" [2] did not use any clustering logic, but just made certain groups in order to teach a system to make better prediction (actually making better fitting functions) in each group. Thus information like distance between objects in X space is not in use at all. Would it become a common rule? Should we proceed in this way, which is a very big departure from the classic clustering, but very close to *intentional analytics*? It seems plausible because, as mentioned, ultimately any classification is to be used not for the sake of classification, but for something else in the future. And this future is "closer than it may appear," especially in our time of Big Brother with Big Data on Big Computers. In [2] the goal was to deliver ads immediately—so, the problem was not about how close objects are to each other in the groups, but how to obtain any "lift" (in monetary sense) versus traditional clustering and/or random distribution.

Question two: is there a way to merge clustering goals with external goals? Or has one to go directly into external criteria not even considering classic clustering anymore?

3. *Clustering in a high-dimensional space*. In seminal work [3], it was proven that the relative difference between maximal and minimal distances goes to one when the dimensionality rises, regardless on the number of observed data points and metric used. This fundamental result was later explored under different angles [4, 5], where some deviations were found (like the one stating that for Manhattan metric the absolute difference between maximal and minimal distances is increasing, etc.), but it doesn't shake the main point—the higher dimension, the less meaningful any distance based clustering is (surprisingly, as experiments show, the effects of non-distinguishability starts at dimensionality (D) as low as 15–20, which is nothing in comparison with some modern problems where D can be 100 or even 1,000).

   It poses an interesting question: why does our intuitively clear perception of "closeness" via distance between objects work extremely well in the familiar three dimensions; work presumably well in somewhat higher dimensions, but does not work at all when D goes big? Does it mean that intuition is wrong and thus even in small dimensions we miss something by relying on distances—or that high dimensions bear something, which cannot be captured by the concept of distance, not just for purely technical reasons (for in high dimensions the degree of sparseness of objects precludes any possibility of these to form clusters)?

It seems that both explanations somehow work. The intuition of the distance goes back to the evolutionary trait of distinguishing food and dangers and definitely has a vital value for humans and other living creatures. But it was developed in the real three-dimensional space, not anywhere else, where all three dimensions are perfectly comparable (not necessarily always equally important). Any mathematical extension beyond this familiar physical world, however attractive it may be, could have little relation to reality—just because we have no way of saying what the "reality" in 100 dimensions is.

By the same token, even if the dimensionality is low, but we try to apply some rigor principles to formulate what we exactly need from the procedure—we face some unsolvable difficulties. It is not a coincidence that no one gave the exact definition of cluster, and all attempts to make definitions "axiomatic" only resulted in some "*impossibility*" statements. One of the first of such results was a theorem by A. Shustorovich (back in 1977, see [1]) about impossibility to create a distance metric satisfying certain logical criteria. One of the recent findings is the theorem by J. Kleinberg about the impossibility of finding a clustering solution satisfying three very logical conditions related to distances [6]. These statements do not depend on dimensionality, but a controversy is apparently embedded in the process of translating intuition into a formal mathematical system. Besides, the irony of any distance calculation in real applications is that here the researcher "in one shot" apparently "solves" the problem, which otherwise would take months: usually, the main scientific goal requires to understand which variables are related to others and this may take a lot of time; here, one takes tens of variables and simply calculates the distance metric—and goes ahead. In the light of that the alternative methods of classification—which do not combine many variables in a single metric—should come forth (and they, of course, started to appear [7]).

Question three: should we correct our typical understanding of clustering procedures given intrinsic flaws in the concept of distance? Should we move to completely distance-free algorithms? Or should we continue using distances as usual in small-dimensional problems and not in high-dimensional (and if children would ask "Why?" reply "Because")?

Methodologically, the current situation in clustering (and, more broadly, in statistics in general) reminds me the situation in modern economics. Historically, the ideas of perfect rationality, equilibrium, self-interest as the only moving force of human behavior, invisible hands of markets, and so on, were so intuitively attractive that played a pivotal role in economics for more than a 100 years. The wonderful theorems were proven, many Nobel Prizes were awarded, and even impossibility theorems like famous ones by K. Arrow did not shake the beautiful building of economics, but just generated a bunch of relaxations and different bypasses. The only thing which undermined its foundation was the stubborn resistance of reality to follow these equations. Remarkably, as was brilliantly shown by P. Mirovski [8], the whole set of concepts in the economics theory was in fact borrowed from physics—but one of the nineteenth century, without the revolution of twentieth. That is,

analogously to clustering, the main metaphor came from the same source—physical intuition—but didn't work well, remaining just a metaphor, not a really working tool.

Only in the last couple of decades, economics slowly started to change. An image of "complexity economics," which captures the deep evolutionary nature of human interactions, emerged [9, 10]. It considers the entire corpus of knowledge about human behavior, economic experiments, agent-based modeling, chaos and catastrophe theories, self-criticality, and other things to capture the evolution driven process. In [11] I proposed the term *Sociosystemics* for the science which would be adequate to reflect these principal changes in our perception of the human relationships (not only those economic ones). This science should cover any socially related issues as an integral whole and require particular approaches to be changed as well.

Is not time come for creation of the "complexity statistics?" And for reconsidering many traditional concepts, such as i.i.d. (because no independent and identically distributed variables actually exist—at least in social life); distance between objects (for all the reasons mentioned); Fisherian theory of the testing of statistical hypotheses (for immense volume of cumulated critiques [12]) and so on? Many of these questions were discussed in [11], where one can find the references.

Does Dr. Mirkin agree with that view? It is my *last but a Big Question*. If yes—there is a hope that all other questions could be resolved as well. The final goal as a driving tool for classification should prevail; question about acceptable and non-acceptable variation in the data will be automatically resolved (via this final goal), and dimensionality curse will be removed thanks to other types of algorithms. Is it not a wonderful subject for a new book?

References

1. Mandel, I.: Cluster Analysis (Klasternyj analiz). Finance and Statistics, Moscow (1988)
2. Chickering, D., Heckerman, D., Meek, C., Platt, J., Thiesson, B.: Goal-oriented clustering. Technical Report MSR-TR-00-82. Microsoft Research Microsoft Corporation. http://research.microsoft.com/pubs/64502/goalclust.pdf (2000)
3. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is 'nearest neighbor' meaningful? In: Proceedings of 7th International Conference on Database Theory (ICDT-1999), Jerusalem, pp. 217–235, 19 (1999)
4. Hinneburg, A., Aggarwal, C., Keim, D.A.: What is the nearest neighbor in high dimensional spaces? In: Proceedings of 26th International Conference on Very Large Data Bases (VLDB-2000), Cairo, September 2000, pp. 506–515
5. Durrant, R., Kabán, A.: When is 'nearest neighbour' meaningful: a converse theorem and implications. J. Complexity **25**(4), 385–397 (2009)
6. Kleinberg, J.: An impossibility theorem for clustering. In: Proceedings of 15th Conference Neural Information Processing Systems, Advances in Neural Information Processing Systems, vol. 15 (2002)
7. Gan, G., Ma, C., Wu, J.: Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia (2007)

8. Mirovski, P.: More Heat than Light: Economics as Social Physics, Physics as Nature's Economics. Cambridge University Press, Cambridge (1989)
9. Beinhocker, E.: Origin of Wealth: Evolution, Complexity, and the Radical Remaking of Economics. Harvard Business School Press, Cambridge (2006)
10. Arthur, W.: Complexity economics: a different framework for economic thought. SFI Working Paper 2013-04-012 (2013)
11. Mandel, I.: Sociosystemics, statistics, decisions. Model Assist. Stat. Appl. **6**, 163–217 (2011)
12. Ziliak, S., McCloskey, D.: The Cult of Statistical Significance: How the Standard Error Costs Us Jobs, Justice, and Lives (Economics, Cognition, and Society). University of Michigan Press, Ann Arbor (2008)

Part I
Classification and Cluster

A Polynomial Algorithm for a Class of 0–1 Fractional Programming Problems Involving Composite Functions, with an Application to Additive Clustering

Pierre Hansen and Christophe Meyer

Abstract We derive conditions on the functions φ, ρ, v and w such that the 0–1 fractional programming problem $\max_{x\in\{0;1\}^n} \frac{\varphi\circ v(x)}{\rho\circ w(x)}$ can be solved in polynomial time by enumerating the breakpoints of the piecewise linear function $\Phi(\lambda) = \max_{x\in\{0;1\}^n} v(x) - \lambda w(x)$ on $[0;+\infty)$. In particular we show that when φ is convex and increasing, ρ is concave, increasing and strictly positive, v and $-w$ are supermodular and either v or w has a monotonicity property, then the 0–1 fractional programming problem can be solved in polynomial time in essentially the same time complexity than to solve the fractional programming problem $\max_{x\in\{0;1\}^n} \frac{v(x)}{w(x)}$, and this even if φ and ρ are non-rational functions provided that it is possible to compare efficiently the value of the objective function at two given points of $\{0;1\}^n$. We apply this result to show that a 0–1 fractional programming problem arising in additive clustering can be solved in polynomial time.

Keywords 0–1 fractional programming • Submodular function • Polynomial algorithm • Composite functions • Additive clustering

1 Introduction

We consider the following 0–1 composite fractional programming problem

$$(CFP) \qquad \max_{x\in B_n} \frac{\varphi \circ v(x)}{\rho \circ w(x)}$$

P. Hansen (✉) • C. Meyer
GERAD, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine,
Montréal, Québec, Canada H3T 2A7
e-mail: pierre.hansen@gerad.ca; christophe.meyer@gerad.ca

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_2,

where $B_n = \{0;1\}^n$, φ and ρ are functions from $\mathbb{R}$ to $\mathbb{R}$ and v and w are functions from B_n to $\mathbb{R}$.

In order for the problem (CFP) to be well-defined, we must assume that $\rho \circ w(x) \neq 0$ for all $x \in B_n$. Actually we will make a stronger assumption and assume that ρ is of constant sign on the convex hull $conv\,(w\,(B_n))$ of the image of B_n by w (we will see later that there is little hope to obtain a polynomial algorithm to solve the problem (CFP) when $\rho \circ w(x)$ can assume both positive and negative values on B_n). More precisely we assume that:

(C1) ρ is strictly positive on $conv\,(w\,(B_n))$.

Since the aim of this paper is to identify polynomial instances of problem (CFP), a natural assumption is:

(C2) evaluation and comparison of the value of the objective function $\frac{\varphi \circ v}{\rho \circ w}$ can be done in polynomial time for any two points x and x' of B_n.

We also need to assume that v and w are rational functions. By redefining φ and ρ if necessary, we assume that

(C3) v and w take integral values on B_n.

We explore a solution approach for problem (CFP) that consists in two steps: first we reduce problem (CFP) to the problem of computing a set of points $X^+ \subseteq B_n$ that define the slopes of the piecewise linear function $\Phi(\lambda) = \max_{x \in B_n} v(x) - \lambda w(x)$ on $[0; +\infty)$; then we consider the problem of computing in an efficient way the set X^+. We show that the reduction step is valid if one of the following sets of assumptions is satisfied:

(C4) there exists $x \in B_n$ such that $(\varphi \circ v)(x) \geq 0$;
(C5) φ and ρ are increasing;
(C6) φ and $-\rho$ are convex;

or:

(C4') $(\varphi \circ v)(x) < 0$ for all $x \in B_n$;
(C5') φ and $-\rho$ are increasing;
(C6') φ and ρ are convex.

Actually we will derive a weaker condition than (C6) and (C6'), but this weaker condition is difficult to exploit as it is expressed in terms of the elements of the set X^+. This weaker condition is implied by (C6) and (C6').

In order for our algorithm to run in polynomial time, we must be able to enumerate in polynomial time the breakpoints of the function Φ. The only nontrivial class of functions that we know for which this can be done in polynomial time is related to the concept of supermodularity. Let us introduce this last set of assumptions:

(C7) v and $-w$ are supermodular on B_n;
(C8) one of the following conditions is satisfied:

(C8a) v or w takes a polynomial number of distinct values on B_n;
(C8b) v and w are both linear;
(C8c) v or w is monotone and the application $x \mapsto (v(x), w(x))$ is weakly bijective on B_n.

The definitions of supermodularity, monotonicity and weak bijection can be found in Sect. 2.1.1. Note that since the opposite of a submodular function is a supermodular function, we could have expressed some of the above assumptions in different equivalent ways. For example, the assumption ("φ is increasing and v is supermodular") is equivalent to the assumption ("φ is decreasing and v is submodular").

Let $T(n)$ be the time to compute the set X^+ and $U(n)$ be the time to evaluate and compare the value of the objective function at two given points x and x' of B_n. The main results of this paper are:

Theorem 1. *If the conditions (C1)–(C8) are satisfied, then problem (CFP) can be solved in polynomial time* $O\left(T(n) + |X^+|U(n)\right)$.

Theorem 2. *If the conditions (C1)–(C3), (C4')–(C6'), (C7) and (C8) are satisfied, then problem (CFP) can be solved in polynomial time* $O\left(T(n) + |X^+|U(n)\right)$.

By polynomial time, we mean a running time that is polynomial in n and in the size of the number $M = \max\left\{\max_{x\in B_n}|v(x)|, \max_{x\in B_n}|w(x)|, \max_{x\in B_n}|(\varphi\circ v)(x)|, \max_{x\in B_n}|(\rho\circ w)(x)|\right\}$.

The remaining of this paper is organized as follows. In Sect. 2 we collect several definitions, facts and results from the literature that are pertinent for our work: the concept of supermodularity is reviewed in Sect. 2.1; Sect. 2.2 is devoted to the minimum cut problem (with non-negative capacities) and to problems reducible to it. In Sect. 2.3 we review in more detail the fractional programming problem with particular emphasis on the so-called Dinkelbach's algorithm.

Section 3 is the main part of this paper. We start by defining more precisely the new algorithm in Sect. 3.1. In Sect. 3.2 we present an algorithm to compute the set X^+ and identify sufficient conditions on the functions v and w that guarantee that this algorithm runs in polynomial time. In Sect. 3.3 we determine conditions on the functions φ and ρ that guarantee that the set X^+ computed by the breakpoint enumeration algorithm of Sect. 3.2 actually contains at least one optimal solution of problem (CFP). Putting together the results of the two previous subsections, we then prove Theorems 1 and 2 in Sect. 3.4, where we also discuss the complexity time of the resulting algorithms.

In Sect. 4 we show how our method can be used to derive a polynomial algorithm for a problem arising in additive clustering.

Extensions of our results to minimization problems, maximization of product of composite functions and constrained problems are discussed in Sect. 5, before we conclude in Sect. 6.

2 Definitions and Related Works

2.1 Supermodularity

2.1.1 Definitions

A function f is *supermodular* over B_n if

$$f(x \wedge y) + f(x \vee y) \geq f(x) + f(y) \qquad \forall x, y \in B_n \tag{1}$$

where $x \wedge y$ is the binary vector whose ith component is the minimum between x_i and y_i, and $x \vee y$ is the binary vector whose ith component is the maximum between x_i and y_i.

A function f is *submodular* if $-f$ is supermodular. A function that is both submodular and supermodular is *modular*. Alternate equivalent definitions exist for super- and submodularity, see, e.g., Nemhauser and Wolsey [41].

For any two vectors x and y of $\mathbb{R}^n$ we write $x \leq y$ if and only if $x_i \leq y_i$ for $i = 1, \ldots, n$, and $x < y$ if and only if $x_i \leq y_i$ for $i = 1, \ldots, n$ and $x \neq y$. We define similarly the notations $x \geq y$ and $x > y$. If neither $x \leq y$ nor $x \geq y$ holds we say that x and y are *not comparable*. Following Topkis [54], a function $f(x)$ from a partially ordered set X to $\mathbb{R}$ is *increasing* (resp. *decreasing*) if $x \leq y$ in X implies $f(x) \leq f(y)$ (resp. $f(x) \geq f(y)$). A function f is *monotone* if it is either increasing or decreasing. A function $f(x)$ from a partially ordered set X to $\mathbb{R}$ is *strictly increasing* (resp. *strictly decreasing*) if $x < y$ in X implies $f(x) < f(y)$ (resp. $f(x) > f(y)$). In this paper the set X will be either $B_n = \{0; 1\}^n$ (a partially ordered set that is not totally ordered) or $\mathbb{R}$ (a partially ordered set that is totally ordered). It is common in lattice theory literature (Topkis [54]) to use the terms *isotone* and *antitone* rather than "increasing" and "decreasing" for a partially ordered set that is not a totally ordered set, but the latter are used herein in order to have a more uniform terminology between the partially ordered set B_n and the totally ordered set $\mathbb{R}$. Although we use only functions with value in a totally ordered set ($\mathbb{R}$ or $\mathbb{N}$), in order to be consistent with Topkis [54] we avoid in this paper the use of the terms "nonincreasing" and "nondecreasing"; in particular the terms "increasing," "decreasing," "strictly increasing" and "strictly decreasing" used in this paper correspond to what may be called "nondecreasing," "nonincreasing," "increasing" and "decreasing" elsewhere.

Finally we say that the application $x \mapsto (v(x), w(x))$ is *weakly bijective* if for all $x, x' \in B_n$,

$$(v(x), w(x)) = \big(v(x'), w(x')\big) \quad \Rightarrow \quad x = x' \text{ or } x \text{ and } x' \text{ are not comparable.}$$

2.1.2 Mathematical Results

The following result, which states some conditions on two functions such that their composition is supermodular or submodular, is due to Topkis [53].

Proposition 1. *Let g be a function defined on B_n, and f be a function defined on $conv\,(g\,(B_n))$, where $conv\,(g\,(B_n))$ denotes the convex hull of the image of set B_n by g.*

a) If f is convex and increasing on $conv\,(g\,(B_n))$ and g is supermodular and monotone on B_n, then $f \circ g$ is supermodular on B_n.
b) If f is convex and decreasing on $conv\,(g\,(B_n))$ and g is submodular and monotone on B_n, then $f \circ g$ is supermodular on B_n.
c) If f is concave and decreasing on $conv\,(g\,(B_n))$ and g is supermodular and monotone on B_n, then $f \circ g$ is submodular on B_n.
d) If f is concave and increasing on $conv\,(g\,(B_n))$ and g is submodular and monotone on B_n, then $f \circ g$ is submodular on B_n.

Proof. We only prove a) since the proof for the other assertions is similar. Let x, y be two elements of B_n. By definition of the operators $\wedge$ and $\vee$, we have $x \wedge y \leq y \leq x \vee y$. Since g is increasing or decreasing, we thus have

$$g(x \wedge y) \leq g(y) \leq g(x \vee y)$$

or

$$g(x \vee y) \leq g(y) \leq g(x \wedge y).$$

In both cases there exists $t \in [0; 1]$ such that

$$g(y) = tg(x \wedge y) + (1-t)g(x \vee y). \tag{2}$$

On the other hand, since g is supermodular and by (2)

$$g(x) \leq g(x \wedge y) + g(x \vee y) - g(y) = tg(x \vee y) + (1-t)g(x \wedge y).$$

Since f is increasing it follows that

$$\begin{aligned} f\left(g(x)\right) &\leq f\left(tg(x \vee y) + (1-t)g(x \wedge y)\right) \\ &\leq tf\left(g(x \vee y)\right) + (1-t)f\left(g(x \wedge y)\right) \\ &= f\left(g(x \vee y)\right) + f\left(g(x \wedge y)\right) - \Big(tf\left(g(x \wedge y)\right) \\ &\quad + (1-t)f\left(g(x \vee y)\right)\Big) \\ &\leq f\left(g(x \vee y)\right) + f\left(g(x \wedge y)\right) - f\left(g(y)\right) \end{aligned}$$

where we used (2) and twice the convexity of f. Hence $f \circ g$ is supermodular on B_n. ■

2.1.3 Supermodular Maximization

Consider the following problem:

$$\text{Supermodular Function Maximization (SFM)} : \max_{x \in B_n} f(x)$$

where f is a supermodular function defined on B_n. Note that SFM could also stand for "Submodular Function Minimization" as for example in [39]. Since a function f is submodular if and only if $-f$ is supermodular and since the problem of maximizing f is equivalent to the problem of minimizing $-f$, the two interpretations are however largely equivalent regarding complexity.

Grötschel, Lovász, and Schrijver [26] were the first to provide a (weakly) polynomial time algorithm for SFM which uses the ellipsoid algorithm for linear programming. It was later shown by the same authors [27] that the ellipsoid algorithm can be used to construct a strongly polynomial algorithm for SFM that runs in $\tilde{O}\left(n^5\text{EO} + n^7\right)$ time. Here the notation $\tilde{O}\left(f(n)\right)$ hides the logarithmic factors, i.e., stands for $O\left(f(n) \cdot (\log n)^k\right)$ for some fixed k and EO stands for the time needed for one evaluation of the objective function. However, this result was not considered very satisfactory since the ellipsoid algorithm is not very practical and does not give much combinatorial insight [39]. Then nearly simultaneously two quite different combinatorial strongly polynomial algorithms (combinatorial in the sense of not using the ellipsoid algorithm) were proposed by Schrijver [49] and Iwata et al. [36], both building on previous works by Cunningham [14]. A few years later Orlin [42] proposed a fully combinatorial strongly polynomial algorithm, i.e., an algorithm that does not use multiplication or division. Let M be an upper bound on $\max_{x \in B_n} |f(x)|$. According to McCormick [39] the best theoretical complexity bounds are $O\left(\left(n^4\text{EO} + n^5\right) \log M\right)$ for weakly polynomial algorithms (Iwata [35]), $O\left(n^5\text{EO} + n^6\right)$ for strongly polynomial algorithms (Orlin [42]) and $O\left(n^8\text{EO} \log^2 n\right)$ for fully combinatorial algorithms (Iwata [35]). See McCormick [39] for a survey of these and other algorithms.

In contrast, maximizing a submodular function is an NP-hard problem as it contains, for example, the *maximum cut* problem in a graph. Therefore, the focus of present works on this problem is to develop good approximation algorithms. This paper does not consider the submodular maximization problem; we refer the reader to [18] for a recent reference.

2.1.4 Parametric Supermodular Maximization: The Notion of Monotone Optimal Solutions

Consider the following parametric supermodular function maximization problem:

$$\text{SFM}(\lambda) \quad \max_{x \in B_n} h(x, \lambda)$$

where λ is either a scalar or a vector of parameters, with value in Λ and $h(x,t)$ is supermodular in x for every $\lambda \in \Lambda$. Let S^*_λ be the set of optimal solutions of problem SFM(λ).

We say that problem SFM(λ) has the *Weak Increasing Optimal Solution Property* (respectively, the *Weak Decreasing Optimal Solution Property*) if for any $\lambda' < \lambda'' \in \Lambda$ and any optimal solution x' of SFM(λ') and any optimal solution x'' of SFM(λ'') it holds that $x' \wedge x''$ (resp. $x' \vee x''$) is an optimal solution of SFM(λ') and $x' \vee x''$ (resp. $x' \wedge x''$) is an optimal solution of SFM(λ'').

The Weak Increasing (resp. Decreasing) Optimal Solution Property implies the existence of an optimal solution x' of SFM(λ') and the existence of an optimal solution x'' of SFM(λ'') such that $x' \leq x''$ (resp. $x' \geq x''$). This ordering relation may, however, not be true for any optimal solutions of SFM(λ') and SFM(λ''). This leads to the definition of the *Strong Increasing Optimal Solution Property* and its decreasing counterpart.

We say that problem SFM(λ) has the *Strong Increasing Optimal Solution Property* (respectively, *Strong Decreasing Optimal Solution Property*) if for any $\lambda' < \lambda'' \in \Lambda$, for any optimal solution x' of SFM(λ') and for any optimal solution x'' of SFM(λ'') it holds that $x' \leq x''$ (resp. $x' \geq x''$).

Finally we say that problem SFM(λ) has the *Weak* (respectively, *Strong*) *Optimal Solution Monotonicity Property* if SFM(λ) has either the *Weak* (resp. *Strong*) *Increasing Optimal Solution Property* or the *Weak* (resp. *Strong*) *Decreasing Optimal Solution Property*.

The Weak and Strong Optimal Solution Monotonicity Property turn out to be a very useful property to prove that some algorithms run in polynomial time, see Proposition 3 together with Propositions 2 and 7.

Sufficient conditions on h have been derived by Topkis [53] (see also [54]) for the problem SFM(λ) to have the Weak Increasing Optimal Solution Property or the Strong Increasing Optimal Solution Property. A straightforward adaptation of his results yields also sufficient conditions for the Weak and Strong Decreasing Optimal Solution Property.

Rather than using these general results, which would require to introduce additional notions, we directly state and prove a sufficient condition for the Weak and Strong Optimal Solution Monotonicity Properties in the particular case where $\Lambda = \{\lambda \in \mathbb{R} : \lambda > 0\}$ and

$$h(x,\lambda) = f(x) - \lambda g(x). \tag{3}$$

A slight improvement can be obtained in this case by replacing the strict monotonicity assumption as a sufficient condition for the Strong Optimal Solution Monotonicity Property by the monotonicity assumption plus the weak bijection property. To see that this is indeed an improvement, consider the pair of functions over B_3: $f(x) = x_1$ and $g(x) = x_2 + x_3$. Both functions are monotone but none of them is strictly monotone. On the other hand, the application $x \mapsto (f(x), g(x))$ is weakly bijective since the only nontrivial solution of equation $(f(x), g(x)) = (f(y), g(y))$ is $x = (u, v, 1-v)$, $y = (u, 1-v, v)$ with $u, v \in \{0; 1\}$

and clearly x and y are not comparable. It is not difficult to show that strict monotonicity implies the weak bijection property.

Proposition 2. *Assume that h has the form (3), where f and $-g$ are supermodular functions on B_n, and that $\Lambda = \{\lambda \in \mathbb{R} : \lambda > 0\}$.*

If f or g is monotone, then $SFM(\lambda)$ has the Weak Optimal Solution Monotonicity Property.

If f or g is monotone and the application $x \mapsto (f(x), g(x))$ is weakly bijective, then $SFM(\lambda)$ has the Strong Optimal Solution Monotonicity Property.

Proof. Let $0 < \lambda' < \lambda''$ and let x' (respectively, x'') be a maximizer of $h(x, \lambda')$ (resp., $h(x, \lambda'')$). We prove the result first in the case where g is increasing, then in the case where f is increasing, and finish by saying a few words on how to modify the proof for the two other cases.

Assume that g is increasing. By optimality of x' and x'',

$$f(x') - \lambda' g(x') \geq f(x' \vee x'') - \lambda' g(x' \vee x'') \tag{4}$$

$$f(x'') - \lambda'' g(x'') \geq f(x' \wedge x'') - \lambda'' g(x' \wedge x''). \tag{5}$$

Summing the two inequalities yields

$$\begin{aligned} & f(x') + f(x'') - f(x' \vee x'') - f(x' \wedge x'') \\ & \geq (\lambda'' - \lambda') \Big(g(x'') - g(x' \wedge x'') \Big) \\ & \qquad + \lambda' \Big(g(x') + g(x'') - g(x' \vee x'') - g(x' \wedge x'') \Big). \end{aligned} \tag{6}$$

The left-hand side of (6) is nonpositive by supermodularity of f while the right-hand side is nonnegative since $\lambda'' > \lambda' \geq 0$ and since g is submodular and increasing (note that $x'' \geq x' \wedge x''$). Hence all inequalities must be satisfied at equality, i.e.,

$$\begin{aligned} f(x') - \lambda' g(x') &= f(x' \vee x'') - \lambda' g(x' \vee x'') \\ f(x'') - \lambda'' g(x'') &= f(x' \wedge x'') - \lambda'' g(x' \wedge x'') \\ f(x') + f(x'') - f(x' \vee x'') - f(x' \wedge x'') &= 0 \\ g(x'') - g(x' \wedge x'') &= 0. \end{aligned}$$

The first two equalities show that $y' = x' \vee x''$ is a maximizer of $h(x, \lambda')$ and $y'' = x' \wedge x''$ is a maximizer of $h(x, \lambda'')$, hence that $SFM(\lambda)$ has the Weak Decreasing Optimal Solution Property. From the remaining equalities it follows that $g(x'') = g(y'')$ and $f(x'') = f(y'')$. Since x'' and y'' are comparable, the weak bijection property implies $x'' = y''$. Since $y'' \leq x'$, we conclude that SFM(λ) has the Strong Decreasing Optimal Solution Property.

Assume now that f is increasing. By multiplying (4) by $\frac{1}{\lambda'}$ and (5) by $\frac{1}{\lambda''}$ and summing, we obtain

$$\begin{aligned} & g(x' \vee x'') + g(x' \wedge x'') - g(x') - g(x'') \\ & \geq -\left(\frac{1}{\lambda'} - \frac{1}{\lambda''}\right)\Big(f(x' \wedge x'') - f(x'')\Big) \\ & \quad + \frac{1}{\lambda'}\Big(f(x' \vee x'') + f(x' \wedge x'') - f(x') - f(x'')\Big). \end{aligned} \tag{7}$$

The left-hand side of (7) is nonpositive by submodularity of g while the right-hand side is nonnegative since $\frac{1}{\lambda'} > \frac{1}{\lambda''} > 0$ and since f is supermodular and increasing. Therefore all inequalities must hold at equality, in particular inequalities (4)–(5). We conclude again that $x' \vee x''$ is a maximizer of $h(x, \lambda')$ and $x' \wedge x''$ is a maximizer of $h(x, \lambda'')$, hence that SFM(λ) has the Weak Decreasing Optimal Solution Property. The Strong Property follows from the monotonicity of f or g and the weak bijection property in the same way than for the case where g is increasing.

If f or g is decreasing we replace inequalities (4)–(5) by

$$\begin{aligned} f(x') - \lambda' g(x') &\geq f(x' \wedge x'') - \lambda' g(x' \wedge x'') \\ f(x'') - \lambda'' g(x'') &\geq f(x' \vee x'') - \lambda'' g(x' \vee x''). \end{aligned}$$

The rest of the proof is similar. In both cases we conclude that SFM(λ) has the Weak or Strong Increasing Optimal Solution Property, depending on whether the weak bijection property holds or not. ■

2.2 *The Minimum Cut Problem*

2.2.1 Definition

Let $G = (V, A)$ be a directed graph with vertex set V and arc set A. With each arc $(v_i, v_j) \in A$ we associate a nonnegative number c_{ij}, called the capacity of the arc (v_i, v_j). Given two subsets S and T of V we denote by (S, T) the set of arcs with origin in S and destination in T, that is

$$(S, T) = \{(v_i, v_j) : v_i \in S \text{ and } v_j \in T\}. \tag{8}$$

Assume that two distinct vertices s and t are given, s being called the *source* and t the *sink*. An (s, t)-*cut*, or more simply a *cut*, is a set $(S, \overline{S})$ (as defined in (8)) with $s \in T$, $t \in \overline{S}$ where $\overline{S} = V \setminus S$ denotes the complement of S. Note that a cut is a set of arcs induced by a set S of nodes. The quantity $c(S, \overline{S}) = \sum_{(v_i, v_j) \in (S, \overline{S})} c_{ij}$ is called the *capacity* of the cut. The *minimum cut problem* consists in determining

the subset S of V that minimizes the capacity $c(S, \overline{S})$. The minimum cut problem can be solved in polynomial time thanks to the max flow–min cut theorem that establishes a strong relation with a linear problem, the *maximum flow problem*, see e.g., Ahuja et al. [1].

2.2.2 The Selection Problem

Hammer and Rudeanu [29] have shown that every function defined on B_n can be written in a unique way as

$$f(x) = \sum_{S \in A} a_S \prod_{i \in S} x_i - \sum_{i=1}^{n} c_i x_i \tag{9}$$

where A is a family of subsets of $\{1, 2, \ldots, n\}$ of size at least 2 and a_S $(S \in A)$ and c_j $(j = 1, \ldots, n)$ are real numbers. An important special case is obtained by adding the restriction

$$a_S \geq 0, \qquad S \in A. \tag{10}$$

When the restriction (10) holds, the problem of maximizing f given by (9) is called a *selection problem* (Rhys [46], Balinski [3]). It was shown by Rhys and Balinski that the selection problem can be formulated as a minimum cut problem in a network defined as follows. With each product of variables $\prod_{i \in S} x_i$ we associate a vertex v_S and with each variable x_i we associate a vertex v_i $(i = 1, \ldots, n)$. There are two more vertices: a source s and a sink t. There is an arc from the source to each vertex v_S with capacity a_S. For each S and for each $i \in S$, there is an arc with infinite capacity from vertex v_S to vertex v_i. Finally for each $i = 1, \ldots, n$ there is an arc from vertex v_i to the sink vertex t with capacity $-c_i$ if $c_i < 0$ or an arc from the source vertex s to the vertex v_i with capacity c_i if $c_i > 0$ (no such arc is needed for vertices v_i such that $c_i = 0$). The network has $n' = |A| + n + 2$ nodes and $m' = |A| + \sum_{S \in A} |S| + n$ arcs.

A network of smaller size exists when the degree of f is ≤ 2 (the degree of f is defined as the largest cardinality of a subset in A). This network has $n + 2$ nodes and $n + |A|$ arcs, see Hammer [34].

It is not difficult to show that the set of functions of degree ≤ 2 that can be written as (9) with the restriction (10) coincides with the set of functions of degree ≤ 2 that are supermodular. This is not true anymore for functions of larger degree as supermodular functions of degree 3 can have negative a_S, see Billionnet and Minoux [5].

2.2.3 The Parametric Minimum Cut Problem

In several applications the capacities of the arcs in a minimum cut problem depend on one or more parameters, and we would like to find a minimum cut for all possible values of the parameters. Gallo et al. [22] have developed an algorithm that solves a special class of parametric minimum cut problem with a single parameter in the same complexity time than what would be necessary to solve the minimum cut problem for a fixed value of the parameter (solving a parametric problem means here to find an optimal solution for all possible values of the parameter). In this special class of parametric minimum cut problem, the capacities of the arcs leaving the source are nondecreasing functions of the (unique) parameter, those of arcs entering the sink are nonincreasing functions of the parameter, and those of all other arcs are constant. The complexity of the Gallo, Grigoriadis and Tarjan algorithm is $O\left(m'n' \log(n'^2/m')\right)$ where n' denotes the number of nodes and m' the number of arcs in the network.

Other classes of the parametric minimum cut problem for which this "all-in-one" property holds have since been identified: see the recent paper by Granot et al. [25] and the references therein.

2.3 *Single-Ratio Fractional Programming*

Problem (CFP) is a 0–1 (single-ratio) fractional programming problem. In general the *(single-ratio) fractional programming* problem is defined as

$$\text{(FP)} \qquad \max_{x \in S} \quad \frac{F(x)}{G(x)} \tag{11}$$

where F and G are real valued functions on a subset S of $\mathbb{R}^n$ and $G(x) > 0$ for all $x \in S$.

The single-ratio fractional programming problem has received considerable attention from the continuous optimization community since the 1960s [10, 16]. According to Frenk and Schaible [19], many of the results on this topic were already presented in the first monograph on fractional programming published in 1978 by Schaible [47]. The focus has since shifted to problems involving multiple ratios, where one, for example, seeks to maximize the sum of several ratios, or maximize the minimum value of several ratios. Other monographs on fractional programming are Craven [13] and Stancu-Minasian [51], see also [20, 48, 52].

The discrete version of the problem also received considerable interests. When $S = \{0; 1\}^n$, the research focused on the case where F and G are polynomials: see, for example, Hansen et al. [31] for the linear case, Hochbaum [33] and the references therein for the quadratic case, Picard and Queyranne [43], Gallo et al. [22] and Chang [9] for polynomials of larger degree.

When constraints are allowed, the functions appearing in the ratio are generally assumed to be linear. Problems that have been considered include the *minimum ratio spanning-tree problem*, the *maximum profit-to-time ratio cycle problem*, the *minimum mean cycle problem*, the *maximum mean cut problem* and the *fractional 0–1 knapsack problem*: see [45] for references to these problems. See also Correa et al. [12], Ursulenko [55].

2.3.1 The Parametric Approach

Almost every solution method developed for fractional programming since the seminal work of Dinkelbach [16] introduces the following auxiliary problem:

$$\text{FPaux}(\lambda) \quad \max_{x \in S} h_\lambda(x) = F(x) - \lambda G(x).$$

λ can be viewed as a "guess" for the optimal value $\frac{F(x^*)}{G(x^*)}$ of problem (FP): if λ is smaller than $\frac{F(x^*)}{G(x^*)}$, the optimal value of the auxiliary problem FPaux(λ) will be positive and its optimal solution will provide a feasible solution with objective value larger than λ; if, on the other hand, λ is larger than $\frac{F(x^*)}{G(x^*)}$, the optimal value of the auxiliary problem will be negative.

We present below Dinkelbach's algorithm for fractional programming. For variants of it, see, e.g., Radzik [45]. Note in particular the proximity of this method with the Newton method for finding roots of polynomial.

DINKELBACH'S ALGORITHM

Step 0. Select some $x^0 \in S$. Compute $\lambda_0 = \frac{F(x^0)}{G(x^0)}$. Set $k = 0$.

Step 1. Solve the auxiliary problem FPaux(λ_k). Let x^{k+1} be an optimal solution.

Step 2. If $h_{\lambda_k}(x^{k+1}) = 0$, stop: $x^* = x^k$. Otherwise let $\lambda_{k+1} = \frac{F(x^{k+1})}{G(x^{k+1})}$, replace k by $k + 1$ and go to Step 1.

The complexity of Dinkelbach's algorithm is determined by the number of iterations and by the complexity of solving the auxiliary problem FPaux(λ) for a given λ. A property that is very useful to derive polynomial algorithms for the fractional programming problem is the supermodularity (see Sect. 2.1).

Consider the 0–1 unconstrained case, i.e., the case where $S = B_n$ and assume that we know that the optimal value of problem (FP) is positive, i.e., that there exists at least one $\tilde{x} \in S$ such that $F(\tilde{x}) \geq 0$. Then we can restrict our attention to $\lambda \geq 0$. If the function $h_\lambda(x)$ is supermodular in x for any $\lambda \geq 0$ then the auxiliary problem FPaux(λ) can generally be solved in polynomial time by one of the algorithms mentioned in Sect. 2.1.3 for SFM. Moreover if FPaux(λ) has the Strong Optimal Solution Monotonicity Property (see Sect. 2.1.4), then the number of iterations in Dinkelbach's algorithm is bounded by n. More precisely we have:

Proposition 3. *Assume that $S = B_n$, that F and G are rational functions that can be evaluated at a given point in polynomial time, that F and $-G$ are supermodular on B_n, that the optimal value of problem (FP) is positive, that either F or G is monotone and that the application $x \mapsto (F(x), G(x))$ is weakly bijective. Then problem (FP) can be solved in polynomial time.*

3 A New Algorithm

This section is the main part of our paper. We start by defining precisely our algorithm in Sect. 3.1. In Sect. 3.2 we characterize the breakpoint vertex set, present an algorithm to compute it and derive conditions on v and w such that this algorithm is polynomial. In Sect. 3.3 we derive conditions on functions φ and ρ that guarantees that our algorithm correctly finds an optimal solution of problem (CFP). Theorems 1 and 2 are proved in Sect. 3.4.

3.1 Description

Let us introduce the function

$$L_\lambda(x) = v(x) - \lambda w(x)$$

and the parametric problem

$$\text{PARAM}(\lambda) \quad \Phi(\lambda) = \max_{x \in B_n} L_\lambda(x).$$

We will denote by $\hat{x}(\lambda)$ an optimal solution of problem PARAM(λ).

Note that the function $L_\lambda(x)$ coincides with the function $h(x, \lambda)$ that would be considered when solving the problem $\max_{x \in B_n} \frac{v(x)}{w(x)}$ by Dinkelbach's algorithm.

It is well-known that $\Phi(\lambda) = \max_{x \in B_n} L_\lambda(x)$ is a convex piecewise linear function on $\mathbb{R}$ (see, e.g., Nemhauser and Wolsey [41, Corollary 6.4]). Let $\mu_1 > \mu_2 > \cdots > \mu_q$ denote the breakpoints of $\Phi(\lambda)$ and let $X = \{x^0, \ldots, x^q\}$ be a subset of B_n such that

$$\Phi(\lambda) = \begin{cases} v(x^q) - \lambda w(x^q), \ \lambda \in (-\infty, \mu_q] \\ v(x^k) - \lambda w(x^k), \ \lambda \in [\mu_{k+1}, \mu_k] \quad \text{for } k = 1, \ldots, q-1 \\ v(x^0) - \lambda w(x^0), \ \lambda \in [\mu_1, +\infty) \end{cases} \tag{12}$$

with

$$w(x^{k-1}) < w(x^k) \qquad k = 1, \ldots, q. \tag{13}$$

By continuity of Φ we have easily

$$\mu_k = \frac{v(x^k) - v(x^{k-1})}{w(x^k) - w(x^{k-1})}, \qquad k = 1, \ldots, q. \tag{14}$$

We will consider subsets of X. Given an interval I of $\mathbb{R}$, we define the set $X_I \subseteq X$ as the set of points x^k needed to define $\Phi(\lambda)$ on the interval I via the formula (12). In particular, $X = X_{(-\infty,+\infty)}$. The set X_I will be called the *breakpoint vertex set* for the function $\Phi(\lambda)$ on interval I. We will be more particularly interested in the set $X_{[0,+\infty)}$, that we will denote more concisely by X^+.

We propose the following algorithm for problem (CFP):

ALGORITHM HM_CFP

Step 1. Construct the set X^+.

Step 2. Compute $x^* = \arg\max_{x \in X^+} \dfrac{(\varphi \circ v)(x)}{(\rho \circ w)(x)}$.

In Sect. 3.2 we study the properties of the set X_I and present an algorithm for its computation. We then determine sufficient conditions on v and w for this algorithm to run in polynomial time in the particular case where $I = [0; +\infty)$. One of the properties identified in Sect. 3.2 is used in Sect. 3.3 to derive conditions on the functions φ and ρ that guarantee the correctness of the algorithm HM_CFP. The results of the previous subsections are used in Sect. 3.4 to identify classes of problems (CFP) that can be solved in polynomial time.

3.2 Computing the Breakpoint Vertex Set

In Sect. 3.2.1 we derive a certain number of properties of the breakpoints and of the breakpoint vertex set, that will be used both to show the correctness of the Eisner and Severance algorithm presented in the next subsection and to derive a sufficient condition on the functions φ and ρ for the set X^+ to contain at least one optimal solution of problem (CFP) in Sect. 3.3. In Sect. 3.2.2 we present the Eisner and Severance algorithm to compute the breakpoint vertex set X_I on a given interval I with at most $2N(I)$ evaluations of the function $\Phi(\lambda)$, where $N(I)$ is the number of breakpoints of Φ on the interval I. Finally in Sect. 3.2.3 we derive conditions on functions v and w that guarantee that the Eisner and Severance algorithm runs in polynomial time when $I = [0; +\infty)$.

3.2.1 Properties

In this section we give some properties of the breakpoint vertex set X introduced in Sect. 3.1. It must be noted that the results in this subsection and in the next one are completely general: no assumption is made on the functions v and w other than being defined on B_n.

We first observe that the set X (and therefore the set X_I for a given interval I) may not be unique if there exists $x', x'' \in B_n$ with $x' \neq x''$ such that $(v(x'), w(x')) = (v(x''), w(x''))$. However both for the purpose of defining the function $\Phi(\lambda)$ and for the algorithm HM_CFP, the two points x' and x'' are completely equivalent. By a slight abuse of language we will continue to write "the set X" (or "the set X_I") in the sequel of this paper.

We now state without proof three easy lemmas.

Lemma 1. *If $\lambda' < \lambda''$ then $-w(\hat{x}(\lambda')) \leq -w(\hat{x}(\lambda''))$.*

Moreover equality holds if and only if $L_{\lambda'}(\hat{x}(\lambda'')) = \Phi(\lambda')$ and $L_{\lambda''}(\hat{x}(\lambda')) = \Phi(\lambda'')$ and in that case we have also $v(\hat{x}(\lambda')) = v(\hat{x}(\lambda''))$.

Lemma 2. *Let $\lambda' < \lambda''$ and assume that $\tilde{x}$ is an optimal solution of both problems* PARAM(λ') *and* PARAM(λ''). *Then $\Phi(\lambda)$ is linear on $[\lambda', \lambda'']$.*

Lemma 3. *$w(x^0) = \min_{x \in B_n} w(x)$. Moreover if there exists more than one optimal solution, x^0 is one of them that maximizes $v(x)$.*

$w(x^q) = \max_{x \in B_n} w(x)$. Moreover if there exists more than one optimal solution, x^q is one of them that maximizes $v(x)$.

The next result will be used to establish sufficient conditions on the functions φ and ρ for the set X^+ to contain an optimal solution of problem (CFP).

Proposition 4. *It holds:*

$$\frac{v(x) - v(x^{k-1})}{w(x) - w(x^{k-1})} \leq \frac{v(x^k) - v(x)}{w(x^k) - w(x)} \qquad \forall x \in B_n : w(x^{k-1}) < w(x) < w(x^k).$$

Proof. By definition of the breakpoints and of the x^k we have

$$v(x^k) - \mu_k w(x^k) \geq v(x) - \mu_k w(x) \qquad \forall x \in B_n.$$

In particular for all $x \in B_n$ such that $w(x^{k-1}) < w(x) < w(x^k)$:

$$\begin{aligned}
& v(x) - \mu_k w(x) \leq v(x^k) - \mu_k w(x^k) \\
\Rightarrow\ & \frac{v(x^k) - v(x^{k-1})}{w(x^k) - w(x^{k-1})} = \mu_k \leq \frac{v(x^k) - v(x)}{w(x^k) - w(x)} \\
\Rightarrow\ & \frac{\Big(v(x^k) - v(x)\Big) + \Big(v(x) - v(x^{k-1})\Big)}{\Big(w(x^k) - w(x)\Big) + \Big(w(x) - w(x^{k-1})\Big)} \leq \frac{\Big(v(x^k) - v(x)\Big)}{\Big(w(x^k) - w(x)\Big)}
\end{aligned}$$

where we used (14). Since each term delimited by a pair of parentheses in the denominators is strictly positive, a simple manipulation gives the announced inequality. ■

We terminate by pointing out that another characterization of the breakpoints can be found in Gallo and Simeone [21], as well as a different approach to compute the breakpoint vertex set that requires to solve a constrained version of problem PARAM(λ).

3.2.2 The Eisner and Severance Algorithm

In this section we present an algorithm to compute the breakpoint vertex set X_I of $\Phi(\lambda)$ on a (finite) interval $I = [\underline{\lambda}, \overline{\lambda}]$, that works for any functions v and w and that requires at most $2N$ solutions of the problem PARAM(λ), where N is the number of breakpoints of $\Phi(\lambda)$ in the interval I. This algorithm was originally proposed by Eisner and Severance [17], see also Gusfield [28].

We first give an informal description of the algorithm. Basically the algorithm partitions the given interval I into subintervals $[\lambda_j, \lambda_{j+1}]$ for $j \in J$. With each λ_j we associate a point $\hat{x}(\lambda_j)$ of B_n that is an optimal solution of problem PARAM(λ_j). The algorithm stops when it can be shown that $\Phi(\lambda)$ is linear on every interval of the partition. Note that by Lemma 2 a sufficient condition for $\Phi(\lambda)$ to be linear on the interval $[\lambda_j, \lambda_{j+1}]$ is that $\hat{x}(\lambda_j)$ is also an optimal solution of problem PARAM(λ_{j+1}) or $\hat{x}(\lambda_{j+1})$ is also an optimal solution of problem PARAM(λ_j). If it is not possible to show that Φ is linear on all intervals of the current partition, we select an interval on which Φ is not known to be linear and subdivide it.

We now explain the subdivision process. Let $[\lambda', \lambda'']$ be an interval to be subdivided, and let $\hat{x}' = \hat{x}(\lambda')$ and $\hat{x}'' = \hat{x}(\lambda'')$ be the optimal solutions associated with the bounds of the interval. We assume that $\hat{x}'$ is not an optimal solution of problem PARAM(λ'') and $\hat{x}''$ is not an optimal solution of problem PARAM(λ') since otherwise Φ would be linear on the interval, which therefore would not have been selected for subdivision. In particular $w(\hat{x}') > w(\hat{x}'')$ by Lemma 1. Define

$$\tilde{\lambda} = \frac{v(\hat{x}') - v(\hat{x}'')}{w(\hat{x}') - w(\hat{x}'')}. \tag{15}$$

We argue that $\tilde{\lambda} \in (\lambda', \lambda'')$. Indeed

$$\tilde{\lambda} - \lambda' = \frac{\Big(v(\hat{x}') - \lambda' w(\hat{x}')\Big) - \Big(v(\hat{x}'') - \lambda' w(\hat{x}'')\Big)}{w(\hat{x}') - w(\hat{x}'')} > 0$$

Step 1 (initialization) : Assume that $I = [\underline{\lambda}, \overline{\lambda}]$. Solve $PARAM(\underline{\lambda})$, obtaining an optimal solution $\hat{x}(\underline{\lambda})$ and an optimal value $\Phi(\underline{\lambda})$. Solve $PARAM(\overline{\lambda})$, obtaining an optimal solution $\hat{x}(\overline{\lambda})$ and an optimal value $\Phi(\overline{\lambda})$. Set $\mathcal{L} = \mathcal{NL} = X_I = \emptyset$ and $\mathcal{N}ew = \{[\underline{\lambda}, \overline{\lambda}]\}$.

Step 2 (linearity test) : For each interval $I' = [\lambda', \lambda'']$ in $\mathcal{N}ew$, do the following:
If $L_{\lambda''}(\hat{x}(\lambda')) = \Phi(\lambda'')$, add $\hat{x}(\lambda')$ to X_I and add I' to $\mathcal{L}$.
If $L_{\lambda'}(\hat{x}(\lambda'')) = \Phi(\lambda')$, add $\hat{x}(\lambda'')$ to X_I and add I' to $\mathcal{L}$.
If none of the above two cases occur, add I' to $\mathcal{NL}$.
Set $\mathcal{N}ew$ to $\emptyset$.

Step 3 (optimality test) : If $\mathcal{NL} = \emptyset$, stop and return X_I.

Step 4 (subdivision) : Select an interval $I' = [\lambda', \lambda'']$ in $\mathcal{NL}$, compute $\tilde{\lambda}$ by the formula (15), solve $PARAM(\tilde{\lambda})$, obtaining an optimal solution $\hat{x}(\tilde{\lambda})$ and an optimal value $\Phi(\tilde{\lambda})$. Set $\mathcal{N}ew = \{[\lambda', \tilde{\lambda}], [\tilde{\lambda}, \lambda'']\}$ and $\mathcal{NL} = \mathcal{NL} \setminus \{I'\}$. Return to Step 2.

Fig. 1 The Eisner and Severance algorithm for computing X_I

since the numerator is strictly positive by optimality of $\hat{x}'$ and non-optimality of $\hat{x}''$ to problem PARAM(λ'). We show in a similar way that $\tilde{\lambda} - \lambda'' < 0$. The subdivision process consists in replacing the interval $[\lambda', \lambda'']$ by the two intervals $[\lambda', \tilde{\lambda}]$ and $[\tilde{\lambda}, \lambda'']$.

The algorithm will maintain three sets: $\mathcal{L}$ is the subset of intervals of the current partition on which Φ was shown to be linear; $\mathcal{NL}$ is the subset of intervals of the current partition on which Φ is not known to be linear and $\mathcal{N}ew$ is the set of new intervals generated during the last iteration. At Step 3, in every iteration, the intervals in $\mathcal{L} \cup \mathcal{NL}$ form a partition of the given interval I. A formal description of the algorithm is given in Fig. 1.

Proposition 5. *Let N be the number of breakpoints of $\Phi(\lambda)$ in interval I, including the lower and upper bounds of I. The above algorithm is correct and terminates after solving at most $2N - 1$ problems* PARAM(λ).

Proof. Consider a point $\hat{x}(t)$ generated by the algorithm, where t is a bound of an interval. There are two possibilities for $\hat{x}(t)$:

- t coincides with a breakpoint. Obviously this case can happen at most N times.
- t lies in the interior of an interval defined by two consecutive breakpoints. To fix the idea assume that $t \in (\mu_k, \mu_{k-1})$ for some k. Then $\hat{x}(t)$ defines the linear piece on $[\mu_k, \mu_{k-1}]$, i.e, $\Phi(\lambda) = v(\hat{x}(t)) - \lambda w(\hat{x}(t))$ for all $\lambda \in [\mu_k, \mu_{k-1}]$. We claim that at most one such point will be generated by the algorithm for each piece of the piecewise linear function $\Phi(\lambda)$. To show this we will prove that if there is another point $\hat{x}(t')$ generated by the algorithm that defines the same linear function, then t' must be a breakpoint. Assume that the values of λ considered by the algorithm are $\lambda_1 < \lambda_2 < \cdots < \lambda_r$. Since we have $w(\hat{x}(\lambda_j)) \geq w(\hat{x}(\lambda_{j+1}))$ for $j = 1, \ldots, r-1$ by Lemma 1 we can assume that $t = \lambda_j$ and $t' = \lambda_{j+1}$ for some j, or the converse. Assume furthermore that λ_j was generated by the algorithm before λ_{j+1} (if this is not the case, we simply invert the roles of λ_j and λ_{j+1}). Then λ_{j+1} corresponds to the $\tilde{\lambda}$ of formula (15) for an interval $[\lambda_j, \lambda'']$, i.e.,

$$\lambda_{j+1} = \frac{v(\hat{x}(\lambda'')) - v(\hat{x}(\lambda_j))}{w(\hat{x}(\lambda'')) - w(\hat{x}(\lambda_j))} = \frac{v(\hat{x}(\lambda'')) - v(\hat{x}(\lambda_{j+1}))}{w(\hat{x}(\lambda'')) - w(\hat{x}(\lambda_{j+1}))}$$

where we used the fact that $v(\hat{x}(\lambda_j)) = v(\hat{x}(\lambda_{j+1}))$ and $w(\hat{x}(\lambda_j)) = w(\hat{x}(\lambda_{j+1}))$ as the two points $\hat{x}(\lambda_j))$ and $\hat{x}(\lambda_{j+1})$ define the same piece of linear function. We then have

$$v(\hat{x}(\lambda_{j+1})) - \lambda_{j+1} w(\hat{x}(\lambda_{j+1})) = v(\hat{x}(\lambda'')) - \lambda_{j+1} w(\hat{x}(\lambda''))$$

which shows that $\hat{x}(\lambda'')$ is an optimal solution of problem PARAM(λ_{j+1}). Hence Φ is linear on $[\lambda_{j+1}, \lambda'']$ by Lemma 2, more precisely $\Phi(\lambda) = v(\hat{x}(\lambda'')) - \lambda w(\hat{x}(\lambda''))$ for $\lambda \in [\lambda_{j+1}, \lambda'']$. Since the interval $[\lambda_j, \lambda'']$ was subdivided, $\hat{x}(\lambda'')$ is not an optimal solution of problem PARAM(λ_j), hence we have $w(\hat{x}(\lambda_j)) > w(\hat{x}(\lambda''))$ by Lemma 1. This shows that the two pieces of linear functions are different on the two intervals $[\lambda_j, \lambda_{j+1}]$ and $[\lambda_{j+1}, \lambda'']$. We therefore conclude that λ_{j+1} is a breakpoint. By the monotonicity of the slopes, there can be no λ_ℓ for $\ell > j+1$ that defines the same linear part than λ_j. We can therefore conclude that the number of generated λ_j that lies strictly between two consecutive breakpoints is bounded by $N - 1$.

Therefore the algorithm generates at most $2N - 1$ points, in particular it is finite. Since the algorithm can only stop when $\mathcal{L}$ contains a partition of the given interval I, we conclude that the algorithm is correct. ■

3.2.3 Complexity

Two conditions must be met for the Eisner and Severance algorithm to compute the set X_I in polynomial time: the number of breakpoints N of $\Phi(\lambda)$ on interval I (or equivalently the size of X_I) must be polynomial in n, and the problem PARAM(λ) must be solvable in polynomial time for fixed λ in I.

In this section we assume that $I = [0; +\infty)$. Note that the upper bound of this interval is not finite as assumed in Sect. 3.2.2, which raises two additional difficulties: we have to find a finite upper bound $\overline{\lambda}$ such that running the Eisner and Severance algorithm on $[0; \overline{\lambda}]$ gives a description of $\Phi(\lambda)$ on the larger interval $[0; +\infty)$, and we have to show that the size of $\overline{\lambda}$ remains polynomial in the size of the data. By (12) and (14), $\overline{\lambda}$ should be chosen such that

$$\overline{\lambda} > \mu_1 = \frac{v(x^1) - v(x^0)}{w(x^1) - w(x^0)}.$$

Let us show that

$$\overline{\lambda} = 1 + \left(\max_{x \in B_n} v(x) \right) - v(\tilde{x})$$

where $\tilde{x}$ is an optimal solution of problem $\min_{x \in B_n} w(x)$, is a valid choice. Since x^0 is an optimal solution of problem $\min_{x \in B_n} w(x)$ that maximizes $v(x)$ by Lemma 3, we have $v(x^0) \geq v(\tilde{x})$. Since $w(x^1) > w(x^0)$ and w takes integral values on B_n we have then $\mu_1 \leq v(x^1) - v(x^0) \leq \left(\max_{x \in B_n} v(x)\right) - v(\tilde{x})$, hence $\overline{\lambda} > \mu_1$. It follows from Sect. 2.1.3 that $\max_{x \in B_n} v(x)$ and $\tilde{x}$ (and therefore $\overline{\lambda}$) can be computed in polynomial time if v is supermodular and w is submodular. Moreover the size of $\overline{\lambda}$ is polynomial in the size of $\max_{x \in B_n} |v(x)|$.

We now consider the condition that problem PARAM(λ) must be solvable in polynomial time for fixed $\lambda \geq 0$. We know of only one sufficiently large class of functions that can be maximized over B_n in polynomial time: it is the class of supermodular functions, see Sect. 2.1. The function $L_\lambda(x)$ is supermodular in x for all $\lambda \geq 0$ if and only if v is supermodular and w is submodular on B_n.

Proposition 6. *If the functions v and $-w$ are supermodular, then the problem* PARAM(λ) *can be solved in polynomial time for any fixed positive λ.*

Proof. Use one of the SFM algorithms mentioned in Sect. 2.1.3. ■

The other necessary condition for the Eisner and Severance algorithm to run in polynomial time is that the set X^+ is of polynomial size. This condition is satisfied in the following cases:

- When the function v or w takes a polynomial number of distinct values on X. Indeed by (13) the sequence $\{w(x^k)\}$ is strictly increasing; and since we restrict ourselves to $\lambda \geq 0$ and by (14), this is also true for the sequence $\{v(x^k)\}$. Thus the number of breakpoints (and hence the size of X^+) is bounded by the number of distinct values taken by v (or w). Examples of such functions are functions that depend on at most $O(\log n)$ variables; $\sum_{j \in J} x_j$ for some subset J of $\{1, 2, \ldots, n\}$; or combination of a fixed number of the above functions.
- When v and w are both linear functions. Indeed it was shown by Hansen et al. [31] that the number of breakpoints is bounded by $n + 1$.
- When PARAM(λ) has the Strong Optimal Solution Monotonicity Property:

  Proposition 7. *Assume that the problem* PARAM(λ) *has the Strong Optimal Solution Monotonicity Property for $\lambda \geq 0$. Then $|X^+| \leq n + 1$.*

  Proof. Let $0 \leq \lambda_1 < \lambda_2 \cdots < \lambda_r$ be the breakpoints generated by the algorithm, sorted in increasing order (i.e., we do not consider here the λ generated by the algorithm that are strictly between two breakpoints). Define $\alpha_i = \frac{\lambda_i + \lambda_{i+1}}{2}$ for $i = 1, \ldots, r-1$, so that each α_i is in the interior of an interval defined by two consecutive breakpoints. If PARAM(λ) has the Strong Optimal Solution Monotonicity Property for $\lambda \geq 0$, then either $\hat{x}(\alpha_1) \leq \hat{x}(\alpha_2) \leq \cdots \leq \hat{x}(\alpha_{r-1})$ or

$\hat{x}(\alpha_1) \geq \hat{x}(\alpha_2) \geq \cdots \geq \hat{x}(\alpha_{r-1})$. Since the $\hat{x}(\alpha_i)$ are all distinct as they define the slopes of the different pieces of the piecewise linear function, we conclude that $r \leq n$. ■

We finally get the following sufficient condition for the algorithm described in Sect. 3.2.2 to run in polynomial time.

Proposition 8. *If the functions v and $-w$ are supermodular and one of the following properties is satisfied:*

- *v or w takes a polynomial number of distinct values on B_n;*
- *v and w are both linear;*
- *v or w is monotone and the application $x \mapsto (v(x), w(x))$ is weakly bijective;*

then the Eisner and Severance algorithm computes the set X^+ in polynomial time.

Proof. Follows from Propositions 2, 6 and 7. ■

3.3 Correctness of the New Algorithm

Let $X = \{x^0, x^1, \ldots, x^q\}$ and X^+ be the sets of points of B_n defined in Sect. 3.1. Let S^* be the set of optimal solutions of problem (CFP). Since $\min_{x \in B_n} w(x) = w(x^0) < w(x^1) < \cdots < w(x^q) = \max_{x \in B_n} w(x)$ by Lemma 3 and (12), for any $x^* \in S^*$ there must exist $k \in \{1, 2, \ldots, q\}$ such that $w(x^{k-1}) \leq w(x^*) \leq w(x^k)$.

The next result considers the case where $w(x^*)$ coincides with the bound of an interval $[w(x^{k-1}), w(x^k)]$, while Proposition 10 considers the case where $w(x^*)$ lies strictly in such an interval. We will assume in this section that φ and ρ are increasing.

Proposition 9. *Assume that φ is increasing and ρ is strictly positive, and let x^* be an optimal solution of problem (CFP). For any $k = 0, \ldots, q$ we have the implication*

$$w(x^*) = w(x^k) \quad \Rightarrow \quad x^k \text{ is an optimal solution of problem (CFP).}$$

Proof. Assume that $w(x^*) = w(x^k)$ for some k. Observe first that $v(x^k) \geq v(x^*)$. Indeed, when $k = 0$, this follows from Lemma 3; when $k \geq 1$, $\Phi(\mu_k) = v(x^k) - \mu_k w(x^k)$ by (12), hence $v(x^k) - \mu_k w(x^k) \geq v(x^*) - \mu_k w(x^*)$. Since $w(x^*) = w(x^k)$, we conclude that $v(x^k) \geq v(x^*)$.

Now since φ is increasing and since $\rho(w(x))$ is strictly positive for all $x \in B_n$, we have easily

$$\frac{\varphi(v(x^k))}{\rho(w(x^k))} \geq \frac{\varphi(v(x^*))}{\rho(w(x^*))}$$

which shows that x^k is an optimal solution of problem (CFP). ■

Proposition 10. *Assume that φ and ρ are increasing, that ρ is strictly positive and that $\max_{x\in B_n}(\varphi\circ v)(x)\geq 0$ and let x^* be an optimal solution of problem (CFP). For any $k=1,\ldots,q$ we have the implication*

$$w(x^{k-1})<w(x^*)<w(x^k)\quad\Rightarrow\quad\begin{cases}v(x^{k-1})<v(x^*)<v(x^k)\\ \qquad\text{or}\\ x^{k-1}\text{is an optimal solution of problem (CFP).}\end{cases}$$

Proof. Assume that $w(x^{k-1})<w(x^*)<w(x^k)$ holds. Since ρ is increasing, we have

$$\rho(w(x^{k-1}))\leq\rho(w(x^*))$$

or, using the fact that ρ is strictly positive,

$$\frac{1}{\rho(w(x^{k-1}))}\geq\frac{1}{\rho(w(x^*))}. \tag{16}$$

We now show that $v(x^*)\leq v(x^{k-1})$ implies that x^{k-1} is an optimal solution of problem (CFP). Assume that $v(x^*)\leq v(x^{k-1})$. The fact that φ is increasing and the assumption on the sign of the optimal value imply that $0\leq\varphi(v(x^*))\leq\varphi(v(x^{k-1}))$. Combining with (16) yields $\frac{\varphi(v(x^{k-1}))}{\rho(w(x^{k-1}))}\geq\frac{\varphi(v(x^*))}{\rho(w(x^*))}$, which shows that x^{k-1} is an optimal solution of problem (CFP). We have thus concluded that either $v(x^*)>v(x^{k-1})$ or x^{k-1} is an optimal solution of problem (CFP).

In the following we assume that the former is true. Since $\Phi(\mu_k)=v(x^{k-1})-\mu_k w(x^{k-1})$ by (12),

$$\begin{aligned}&v(x^{k-1})-\mu_k w(x^{k-1})\geq v(x^*)-\mu_k w(x^*)\\ \Rightarrow\ &\mu_k\left(w(x^*)-w(x^{k-1})\right)\geq v(x^*)-v(x^{k-1})>0.\end{aligned}$$

Since $w(x^*)>w(x^{k-1})$ we conclude that $\mu_k>0$. Now since we have also $\Phi(\mu_k)=v(x^k)-\mu_k w(x^k)$,

$$\begin{aligned}&v(x^k)-\mu_k w(x^k)\geq v(x^*)-\mu_k w(x^*)\\ \Rightarrow\ &v(x^*)-v(x^k)\leq\mu_k\left(w(x^*)-w(x^k)\right)<0.\end{aligned}$$

Hence $v(x^{k-1})<v(x^*)<v(x^k)$. ■

Propositions 9 and 10 leave open the possibility that $x^*=x^k$ for some k such that $\mu_k<0$. The next result shows that if that happens, then at least one x^ℓ with ℓ such that $\mu_\ell>0$ is also an optimal solution of problem (CFP).

Proposition 11. *Assume that φ and ρ are increasing, that ρ is strictly positive and that* $\max_{x\in B_n}(\varphi\circ v)(x)\geq 0$. *If $S^*\cap X\neq\emptyset$ then $S^*\cap X^+\neq\emptyset$.*

Proof. Recall that by definition $\mu_1>\mu_2>\cdots>\mu_q$. If $\mu_q\geq 0$ we are done, so assume that $\mu_q<0$. Define r to be such that $\mu_{r-1}\geq 0>\mu_r$. Then we have $\mu_k<0$ for all $k=r,\ldots,q$. Since $w(x^k)>w(x^{k-1})$ for all k and by (14), it follows that

$$v(x^k)<v(x^{k-1}),\qquad k=r,\ldots,q. \tag{17}$$

Now assume that x^t is an optimal solution of problem (CFP) with $r\leq t\leq q$. We will show that x^t is also an optimal solution of problem (CFP). Since the sequence $\{w(x^k)\}$ is strictly increasing and by (17)

$$v(x^t)<v(x^r)$$
$$w(x^t)>w(x^r).$$

Since φ and ρ are increasing and ρ is strictly positive we get

$$(\varphi\circ v)(x^t)\leq(\varphi\circ v)(x^r)$$
$$0<\frac{1}{(\rho\circ w)(x^t)}\leq\frac{1}{(\rho\circ w)(x^r)}.$$

Hence, since $(\varphi\circ v)(x^t)\geq 0$,

$$\frac{(\varphi\circ v)(x^t)}{(\rho\circ w)(x^t)}\leq\frac{(\varphi\circ v)(x^r)}{(\rho\circ w)(x^r)}.$$

Therefore x^r is also an optimal solution of problem (CFP), and x^r belongs to X^+ as it defines $\Phi(t)$ over the interval $[0;\mu_{r-1}]$. ■

The next result establishes a sufficient condition on φ and ρ for the existence of an optimal solution of problem (CFP) in the set X^+.

Proposition 12. *Assume that φ and ρ are increasing and that* $\max_{x\in B_n}\varphi(v(x))\geq 0$. *A sufficient condition for $S^*\cap X^+\neq\emptyset$ is*

$$\left(\frac{t-v(x^{k-1})}{\varphi(t)-\varphi(v(x^{k-1}))}\right)\left(\frac{\rho(u)-\rho(w(x^{k-1}))}{u-w(x^{k-1})}\right)\left(\frac{\varphi(v(x^k))-\varphi(t)}{v(x^k)-t}\right)$$
$$\times\left(\frac{w(x^k)-u}{\rho(w(x^k))-\rho(u)}\right)\geq 1$$
$$\forall t: v(x^{k-1})<t<v(x^k),\quad \forall u: w(x^{k-1})<u<w(x^k) \tag{18}$$

Proof. We will assume that $S^*\cap X^+=\emptyset$ and exhibit a couple (t,u) that violates (18).

Let $x^* \in S^*$. By Propositions 9 and 11, $w(x^*) \neq w(x^k)$ for all $k = 0, 1, \ldots, q$. Therefore since $w(x^0) = \min_{x \in B_n} w(x)$ and $w(x^q) = \max_{x \in B_n} w(x)$ by Lemma 3, there must exist some k such that $w(x^*) \in (w(x^{k-1}), w(x^k))$.

To simplify the notations, let $v^* = v(x^*)$, $w^* = w(x^*)$, $v_\ell = v(x^\ell)$ and $w_\ell = w(x^\ell)$ for $\ell \in \{k-1, k\}$. Since $x^{k-1} \notin S^*$ and by Propositions 10 and 11 we have

$$v_{k-1} < v^* < v_k. \tag{19}$$

By optimality of x^* and since $x^{k-1}, x^k \notin S^*$,

$$\frac{\varphi(v^*)}{\rho(w^*)} > \frac{\varphi(v_{k-1})}{\rho(w_{k-1})} \tag{20}$$

$$\frac{\varphi(v^*)}{\rho(w^*)} > \frac{\varphi(v_k)}{\rho(w_k)}. \tag{21}$$

Now by (21)

$$\varphi(v_k) - \varphi(v^*) < \frac{\varphi(v^*)}{\rho(w^*)} \left(\rho(w_k) - \rho(w^*)\right). \tag{22}$$

Since $v_k > v^*$ and φ is increasing, we cannot have $\varphi(v^*) = 0$ hence it follows from the assumptions that $\varphi(v^*) > 0$. Therefore the fact that φ is increasing implies that $\rho(w_k) - \rho(w^*) > 0$. Inequality (22) can then be written:

$$\frac{\varphi(v^*)}{\rho(w^*)} > \frac{\varphi(v_k) - \varphi(v^*)}{\rho(w_k) - \rho(w^*)}. \tag{23}$$

Similarly (20) yields

$$\varphi(v^*) - \varphi(v_{k-1}) > \frac{\varphi(v^*)}{\rho(w^*)} \left(\rho(w^*) - \rho(w_{k-1})\right). \tag{24}$$

Since $w^* > w_{k-1}$, ρ is increasing and $\varphi(v^*) > 0$, it follows that $\varphi(v^*) - \varphi(v_{k-1}) > 0$. Hence (24) can be written

$$\frac{\rho(w^*)}{\varphi(v^*)} > \frac{\rho(w^*) - \rho(w_{k-1})}{\varphi(v^*) - \varphi(v_{k-1})}. \tag{25}$$

By Proposition 4

$$\frac{v^* - v_{k-1}}{w^* - w_{k-1}} \leq \frac{v_k - v^*}{w_k - w^*}$$

$$\Rightarrow 1 \geq \left(\frac{v^* - v_{k-1}}{w^* - w_{k-1}}\right)\left(\frac{w_k - w^*}{v_k - v^*}\right). \tag{26}$$

Since each factor in inequalities (23), (25) and (26) is nonnegative, we can multiply these two inequalities memberwise, hence

$$\left(\frac{v^* - v_{k-1}}{\varphi(v^*) - \varphi(v_{k-1})}\right)\left(\frac{\rho(w^*) - \rho(w_{k-1})}{w^* - w_{k-1}}\right)\left(\frac{\varphi(v_k) - \varphi(v^*)}{v_k - v^*}\right)\left(\frac{w_k - w^*}{\rho(w_k) - \rho(w^*)}\right) < 1.$$

We have shown that the assumption $S^* \cap X^+ = \emptyset$ implies that (18) is violated for $t = v^*$ and $u = w^*$. Hence (18) implies that $S^* \cap X^+ \neq \emptyset$. ■

In order to derive a simpler condition on φ and ρ that implies the sufficient condition of Proposition 12, we need the following Lemma.

Lemma 4. *Let h be a convex function over an interval $[a, b]$. Then*

$$\frac{h(t) - h(a)}{t - a} \leq \frac{h(b) - h(t)}{b - t} \qquad \forall t : a < t < b. \tag{27}$$

Proposition 13. *Assume that φ and ρ are increasing. If in addition φ is convex and ρ is concave, then the sufficient condition (18) for $S^* \cap X^+ \neq \emptyset$ is satisfied.*

Proof. Let (t, u) such that $v(x^{k-1}) < t < v(x^k)$ and $w(x^{k-1}) < u < w(x^k)$. If φ is convex and ρ is concave, we have by Lemma 4

$$\frac{\varphi(t) - \varphi(v(x^{k-1}))}{t - v(x^{k-1})} \leq \frac{\varphi(v(x^k)) - \varphi(t)}{v(x^k) - t} \tag{28}$$

$$\frac{\rho(u) - \rho(w(x^{k-1}))}{u - w(x^{k-1})} \geq \frac{\rho(w(x^k)) - \rho(u)}{w(x^k) - u}. \tag{29}$$

Since φ and ρ are increasing, each ratio in these two inequalities is strictly positive. We then easily derive (18). ■

To conclude this section we give an example that shows that the assumption of convexity is not necessary for (27) to be satisfied in Lemma 4. This in particular implies that the sufficient condition of Proposition 12 can be satisfied even when φ is not convex and ρ is not concave.

Example 1. Consider the following piecewise linear function defined on the interval $[a, b] = [0; 5]$.

$$h(t) = \begin{cases} t & \text{if } t \in [0; 2] \\ 6t - 10 & \text{if } t \in [2; 3] \\ 2t + 2 & \text{if } t \in [3; 5]. \end{cases}$$

The graph of this function is represented in Fig. 2. Clearly $h(t)$ is not convex. It can be verified that for any position of the point T on the curve, the slope of the segment AT is smaller than the slope of the segment TB. This is exactly what is expressed by (27).

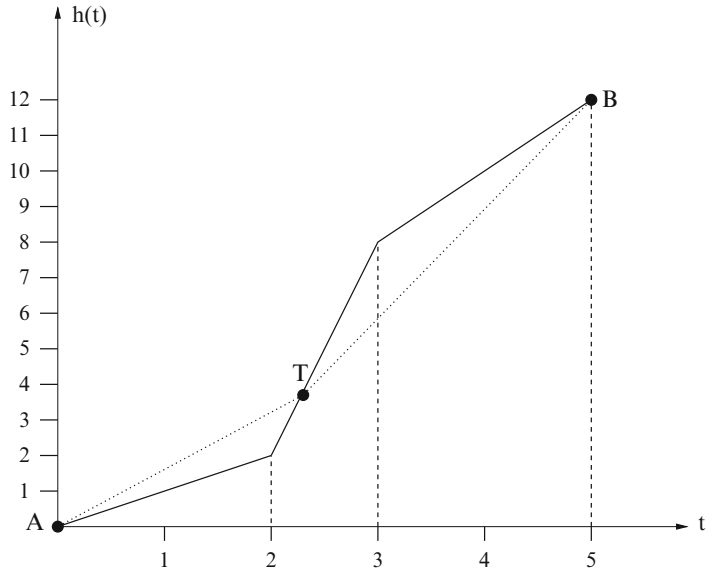


Fig. 2 Geometrical interpretation of the inequality of Lemma 4

3.4 Polynomial Solvable Instances

In this section we prove Theorems 1 and 2.

Proof of Theorem 1 Follows from Propositions 8, 12 and 13. ∎

Proof of Theorem 2 Assume that the condition (C4') is satisfied, i.e., $(\varphi \circ v)(x) < 0$ for all $x \in B_n$. We observe that

$$\max_{x \in B_n} \frac{(\varphi \circ v)(x)}{(\rho \circ w)(x)} \Leftrightarrow \min_{x \in B_n} \frac{-(\varphi \circ v)(x)}{(\rho \circ w)(x)} \Leftrightarrow \max_{x \in B_n} \frac{(\rho \circ w)(x)}{-(\varphi \circ v)(x)}$$
$$\Leftrightarrow \max_{x \in B_n} \frac{(\varphi' \circ v')(x)}{(\rho' \circ w')(x)}$$

with $\varphi'(t) = \rho(-t)$, $\rho'(t) = -\varphi(-t)$ for $t \in \mathbb{R}$ and $v'(x) = -w(x)$ and $w'(x) = -v(x)$ for all $x \in B_n$. The result then follows from Theorem 1 applied to the problem $\max_{x \in B_n} \frac{(\varphi' \circ v')(x)}{(\rho' \circ w')(x)}$ and by converting the conditions on φ', ρ', v' and w' to conditions on φ, ρ, v and w. ∎

The equivalent of Proposition 12 for the instances satisfying condition (C4') is:

Proposition 14. *Assume that φ is increasing, that ρ is decreasing and that $\max_{x \in B_n} \varphi(v(x)) < 0$. A sufficient condition for $S^* \cap X^+ \neq \emptyset$ is*

$$\left(\frac{t - v(x^{k-1})}{\varphi(t) - \varphi(v(x^{k-1}))}\right)\left(\frac{\rho(u) - \rho(w(x^{k-1}))}{u - w(x^{k-1})}\right)\left(\frac{\varphi(v(x^k)) - \varphi(t)}{v(x^k) - t}\right)\left(\frac{w(x^k) - u}{\rho(w(x^k)) - \rho(u)}\right) \geq 1$$
$$\forall t : v(x^{k-1}) < t < v(x^k), \quad \forall u : w(x^{k-1}) < u < w(x^k). \tag{30}$$

In particular, the inequality (30) coincides with (18).

We terminate with some remarks:

- In view of Proposition 5, it is natural to assume that the time $T(n)$ to compute the set X^+ is of the order of $|X^+|V(n)$ where $V(n)$ is the time needed to solve the problem PARAM(λ) for some given λ. Actually this can often be done in the order of $V(n)$ by using a different algorithm, see, for example, Sect. 2.2.3.
- The complexity of our algorithm is essentially determined by the computation of the set X^+, which depends only upon the functions v and w. The lowest complexities will be obtained when v and w can be represented by low degree multilinear polynomials. For example, if v and w are both linear, it is possible to compute the set X^+ in $O(n \log n)$ by representing implicitly the elements of X^+. If v and $-w$ are quadratic supermodular functions and some monotonicity property holds, the algorithm of Gallo, Grigoriadis and Tarjan can compute the set X^+ in $O(n^6)$, see Sects. 2.2.2 and 2.2.3.
- Conversely, since the functions φ and ρ are used only to identify the best point in X^+, they can have less attractive properties, for example—they could be non-rational as illustrated by the additive clustering problem, see Sect. 4.3.
- If an instance (φ, ρ, v, w) satisfies the assumptions of Theorem 1, one must in particular have that φ is increasing (condition (C5)) and convex (condition (C6)) and v is supermodular (condition (C7)). By part a) of Proposition 1, only the absence of the monotonicity property for v prevents us from concluding that $\varphi \circ v$ is supermodular (and monotone) on B_n. Similarly, by part d) of Proposition 1, only the absence of the monotonicity property for w prevents us from concluding that $\rho \circ w$ is submodular (and monotone) on B_n. If both v and w were monotone, the assumptions of Theorem 1 are thus close to allow a polynomial algorithm for (CFP) via the direct use of Dinkelbach's algorithm: we would need in addition that φ and ρ are rational functions, and a kind of strict monotonicity for either $\varphi \circ v$ or $\rho \circ w$ but these additional assumptions are relatively minor with respect to those of Theorem 1. One can even notice that monotonicity of v and w is present in the assumption of Theorem 1 (condition (C8c)). This suggests that our new class of polynomially solvable instances extends only marginally the known class of polynomially solvable instances of the fractional programming problem.

  This is not exactly true, because our results do not require either v or w to be monotone: condition (C8) could be satisfied through either (C8a) or (C8b). And even if condition (C8c) is satisfied, only one of the functions v or w need to be monotone. In other words $\varphi \circ v$ and/or $-\rho \circ w$ might not have the supermodularity property when the assumptions of Theorem 1 or 2 are satisfied, in which case we do not know how to solve efficiently the auxiliary problem that arises in Dinkelbach's method.

4 Application to an Additive Clustering Problem

In this section we show how the results obtained up to now can be used to derive a polynomial algorithm for a problem arising in additive clustering. We start by introducing additive clustering in Sect. 4.1. In Sect. 4.2 we reformulate a particular problem arising in this area as a 0–1 fractional programming problem. An $O(n^5)$ algorithm to solve this later problem is then described in Sect. 4.3.

4.1 Additive Clustering

The additive clustering (ADCLUS) model has been introduced by Shepard and Arabie [50], Arabie and Carroll [2] in the context of cognitive modeling.[1] This model assumes that the similarity between two objects, measured by a nonnegative number, is additively caused by the properties (also called *features*) that these two objects share. With each property we can associate a cluster, which contains all objects that have this property. Furthermore with each cluster we associate a positive weight representing the importance of the corresponding property. The similarity predicted by the model for a pair of objects is then defined as the sum of the weights of the clusters to which both objects belong (note that clusters can overlap). An ADCLUS model is characterized by a set of clusters, together with their weights. Given a similarity matrix obtained typically by some experiments, the additive clustering problem consists in constructing a model that explains as much as possible of the given similarity matrix, under the restriction that the model's complexity is limited (if we do not restrict the complexity of the model, we can reconstruct perfectly the similarity matrix with $O(n^2)$ clusters, see Shepard and Arabie [50, p. 98]). We will assume here that the complexity of the model is measured by the number of clusters, see, e.g., Lee and Navarro [38] and references therein for more elaborated measures of the complexity. In other words, limiting the complexity of the model amounts to setting an upper bound on the number of clusters used to construct the approximate similarity matrix. Many authors have developed algorithms to fit this model (or variants or generalizations of it) with this definition of the complexity, see, e.g., [4, 7, 11, 15, 37, 40].

The mathematical formulation of the additive clustering problem with m clusters is the following:

$$\text{ADCLUS}(m) \quad \min \quad f(x,w) = \sum_{i<j} \left(s_{ij} - \sum_{k=1}^{m} w_k x_i^k x_j^k \right)^2$$

[1] A similar model was developed independently and at the same time in the former USSR; see Mirkin [40] and the references therein.

$$\begin{array}{lll} \text{s.t.} & w_k \geq 0 & k = 1, \ldots, m \\ & x_i^k \in \{0,1\} & k = 1, \ldots, m;\ i = 1, \ldots, n \end{array}$$

where $S = (s_{ij})$ is a $n \times n$ symmetrical nonnegative matrix.

4.2 The Additive Clustering Problem with One Cluster

In an attempt to assess the complexity of problem ADCLUS(m) we studied the version with one cluster:

$$\begin{array}{lll} \text{ADCLUS(1)} & \min & f(x,w) = \sum_{i<j} \left(s_{ij} - w x_i x_j\right)^2 \\ & \text{s.t.} & w \geq 0 \\ & & x_i \in \{0,1\} \qquad i = 1, \ldots, n. \end{array}$$

Note that the cluster must have at least two elements in order to define a non-null reconstructed matrix. This motivates the introduction of the set T:

$$T = \left\{ x \in B_n : \sum_{i=1}^{n} x_i \geq 2 \right\}. \tag{31}$$

We now reformulate problem ADCLUS(1) as a 0–1 fractional programming problem.

Proposition 15. *Problem ADCLUS(1) is equivalent to*

$$\text{ADCLUS}'(1) \qquad \max_{x \in T} \quad g(x) = \frac{\left(\sum_{i<j} s_{ij} x_i x_j \right)^2}{\left(\sum_{i=1}^{n} x_i \right) \left(\sum_{i=1}^{n} x_i - 1 \right)}.$$

In particular if x^* is an optimal solution of problem ADCLUS$'(1)$ *then (x^*, w^*) is an optimal solution of problem ADCLUS(1) with*

$$w^* = \frac{\sum_{i<j} s_{ij} x_i^* x_j^*}{\sum_{i<j} x_i^* x_j^*}.$$

Proof. See Hansen et al. [32]. ■

4.3 A Polynomial Algorithm

We first explain how a straightforward application of the results of this paper lead to an $O(n^5)$ algorithm to solve problem ADCLUS$'(1)$. Then we show that with a little additional effort an $O(n^4)$ algorithm can be obtained.

Problem ADCLUS$'(1)$ is not a (CFP) problem because its feasible set is a strict subset of B_n. However note that problem ADCLUS$'(1)$ can be reduced to a polynomial number of problems (CFP) of size $n-2$, each problem being obtained from ADCLUS$'(1)$ by fixing two variables to 1. By renumbering the variables if necessary, the general form of such a problem is

$$\text{ADCLUS}'_2(1) \qquad \max_{x\in B_{n-2}} \tilde{g}_2(x) = \frac{\left(\sum_{i<j} \tilde{s}_{ij}x_ix_j + \sum_{i=1}^{n-2} \tilde{s}_{ii}x_i + \tilde{c}\right)^2}{\left(\sum_{i=1}^{n-2} x_i + 2\right)\left(\sum_{i=1}^{n-2} x_i + 1\right)}$$

where $\tilde{S}$ is a $(n-2)\times(n-2)$ symmetrical matrix with nonnegative entries and $\tilde{c}$ is a nonnegative constant. Clearly solving problem ADCLUS$'(1)$ in polynomial time is equivalent to solving problem ADCLUS$'_2(1)$ in polynomial time.

Unfortunately problem ADCLUS$'_2(1)$ does not satisfy the assumptions of Theorem 1 or 2, so we consider instead the problem:

$$\text{ADCLUS}''_2(1) \qquad \max_{x\in B_{n-2}} \tilde{h}_2(x) = \frac{\sum_{i<j} \tilde{s}_{ij}x_ix_j + \sum_{i=1}^{n-2} \tilde{s}_{ii}x_i + \tilde{c}}{\sqrt{\left(\sum_{i=1}^{n-2} x_i + 2\right)\left(\sum_{i=1}^{n-2} x_i + 1\right)}}$$

Since the matrix S is assumed to be nonnegative, the numerator $\sum_{i<j} \tilde{s}_{ij}x_ix_j + \sum_{i=1}^{n-2} \tilde{s}_{ii}x_i + \tilde{c}$ is nonnegative for all $x \in B_{n-2}$, hence problem ADCLUS$'_2(1)$ is equivalent to problem ADCLUS$''_2(1)$. Now problem ADCLUS$''_2(1)$ is a problem (CFP) with $\varphi = \tilde{\varphi}$, $\rho = \tilde{\rho}$, $v = \tilde{v}$ and $w = \tilde{w}$ where

$$\begin{aligned}
\tilde{\varphi}(t) &= t \\
\tilde{\rho}(t) &= \sqrt{(t+1)(t+2)} \\
\tilde{v}(x) &= \sum_{i<j} \tilde{s}_{ij}x_ix_j + \sum_{i=1}^{n-2} \tilde{s}_{ii}x_i + \tilde{c} \\
\tilde{w}(x) &= \sum_{i=1}^{n-2} x_i
\end{aligned}$$

and it can be verified that $(\tilde{\varphi}, \tilde{\rho}, \tilde{v}, \tilde{w})$ satisfies the conditions of Theorem 1.

The corresponding problem PARAM(λ) can be reformulated as a parametric minimum cut problem in a network with n vertices and $O(n^2)$ arcs, which can be solved by the Gallo, Grigoriadis and Gallo algorithm, see Sect. 2.2.3. Hence the time needed to compute the set X^+ is $T(n) = O(n^3)$. Step 2 of the HM_CFP algorithm consists in identifying the best feasible point among the points computed in Step 1. In order to avoid problems with the square-root function, we evaluate $\left(\tilde{h}_2(x)\right)^2$ instead of $\tilde{h}_2(x)$. An evaluation costs $O(n^2)$ time, hence the complexity of Step 2 is in $O(n^3)$. Thus the overall complexity for solving ADCLUS$'_2(1)$ is $O(n^3)$. Since we need to solve $O(n^2)$ of such problems to solve problem ADCLUS(1), the complexity of this latter problem is $O(n^5)$. Hence we have shown:

Proposition 16. *There exists an $O(n^5)$ algorithm to solve problem ADCLUS(1).*

Proposition 16 suffices to show that problem ADCLUS(1) can be solved in polynomial time. The question is now whether we can lower the order of the complexity. A little attention shows that we can obtain an $O(n^4)$ algorithm by working with problems obtained by fixing one variable to 1 rather than two. The resulting problems are problems (CFP) with B_n replaced by $B_n \setminus \{(0)\}$. By looking at the proofs, we observe that the analysis made for the unconstrained case remains valid, hence yielding an $O(n^4)$ algorithm. We believe that this complexity can still be improved but let this be for further research. Let us only note that working directly with problem ADCLUS$'(1)$ by setting $\rho(t) = \sqrt{t(t-1)}$ does not work, as shown by the following example.

Example 2. Let $n = 8$ and consider the following instance of ADCLUS$'(1)$ where we maximize the square-root of the original objective function:

$$\max \quad \frac{18x_1x_5 + 15x_1x_6 + 8x_2x_7 + 16x_3x_5 + 22x_4x_6 + 15x_6x_8}{\sqrt{\left(\sum_{i=1}^{8} x_i\right)\left(\sum_{i=1}^{8} x_i - 1\right)}}$$

$$\text{s.t.} \quad \sum_{i=1}^{8} x_i \geq 2$$

$$x_i \in \{0;1\} \qquad i = 1, \ldots, 8.$$

Then

$$\Phi(\lambda) = \max_{x \in B_8} \left\{ 18x_1x_5 + 15x_1x_6 + 8x_2x_7 + 16x_3x_5 + 22x_4x_6 + 15x_6x_8 - \lambda \left(\sum_{i=1}^{8} x_i\right) \right\}.$$

Applying the Eisner and Severance algorithm yields

$$\Phi(\lambda) = \begin{cases} 94 - 8\lambda & 0 \le \lambda \le 4 \\ 86 - 6\lambda & 4 \le \lambda \le 14.33 \\ 0 & 14.33 \le \lambda \end{cases}$$

with $X^+ = \{(11111111), (10111101), (00000000)\}$. The best point among these three can easily be shown to be (10111101) with a value for the (squared-rooted) objective function equal approximately to 11.5962. However the optimal solution of problem ADCLUS$'(1)$ is $x^* = (10011100)$ with an (square-rooted) objective value of approximately 15.8771.

As observed in Sect. 3.4, all instances of problem (CFP) solvable by our method do not satisfy the property that $\varphi \circ v$ and $-\rho \circ w$ are monotone supermodular functions, but it turns out that this is true for problem ADCLUS$''_2(1)$. Thus it seems that Dinkelbach's algorithm would be polynomial too. In fact the only assumption of Proposition 3 that is not satisfied is that $x \mapsto \sqrt{\left(\sum_{i=1}^{n-2} x_i + 2\right)\left(\sum_{i=1}^{n-2} x_i + 1\right)}$ is a rational function. We now discuss why the non-rationality of this function is a serious difficulty if we want to show that Dinkelbach's algorithm is polynomial. Recall that Dinkelbach's algorithm requires the solution of the following auxiliary problem

$$\text{FPaux}_{1/2}(\lambda) \quad \max_{x \in B_{n-2}} h_{\lambda,1/2}(x) = \sum_{i<j} \tilde{s}_{ij} x_i x_j + \sum_{i=1}^{n-2} \tilde{s}_{ii} x_i + \tilde{c} - \lambda \sqrt{\left(\sum_{i=1}^{n-2} x_i + 2\right)\left(\sum_{i=1}^{n-2} x_i + 1\right)}$$

for some λ. This auxiliary problem is solved typically by one of the SFM algorithms mentioned in Sect. 2.1.3. Such an algorithm requires from time to time the evaluation of the objective function $h_{\lambda,1/2}$ at some points of B_{n-2}. The question that arises is what is the precision needed for λ and for the evaluation of the objective function value at a point of B_{n-2} in order to guarantee that the solution returned by the SFM algorithm is indeed optimal (note that from the point of view of the correctness of Dinkelbach's algorithm, the optimality of the solution returned by the SFM algorithm is required only at the last iteration; however, if the SFM algorithm returns non-optimal solutions at other iterations, the number of iterations might not anymore be polynomial)? Let us approach this question differently. It is easy to see that the λ_k are of the form $b\sqrt{c}$ where b is a rational and c is an integer of the form $(t+1)(t+2)$ with $t \in \{0, \ldots, n\}$, hence the value of the objective function $h_{\lambda,1/2}(x)$ can be written as $a' + b'\sqrt{c'}$ where c' is a square-free integer less than $n^2(n+1)^2$, in particular the objective function value can be outputed exactly by specifying the triple (a', b', c'). Now these values are likely to be added, subtracted and compared together by the SFM algorithm (if we restrict ourselves to a fully strongly combinatorial algorithm we do not have to care about multiplication and division). To represent exactly a sum of such numbers, we can introduce a basis that spans the set of numbers $\bigcup_{1 \le c' \le n^2(n+1)^2} \{\sqrt{c'}\}$. The question is

now: what is the complexity of comparing two numbers written in this basis? Up to now, no polynomial algorithm is known for this comparison problem [6, 44]. It is not impossible that a finer analysis, taking into account what operations exactly are done on these numbers by the SFM algorithm as well as the structure of the numbers forming the basis, would yield a polynomial algorithm by this approach, but this might not be easy. Even if it is possible, the best complexity we could hope for the problem ADCLUS(1) by applying Dinkelbach's algorithm in conjunction with a generic SFM algorithm is $O(n^9)$ if we use Orlin's strongly polynomial algorithm (that uses multiplication and division) or $O(n^{12}\log^2 n)$ if we use Iwata's fully strongly combinatorial algorithm, which is much higher than the $O(n^5)$ algorithm described in this section.

5 Discussions: Limitations and Extensions

In the previous section we have shown that the problem (CFP) can be solved in polynomial time if the functions φ, ρ, v and w satisfy the conditions of Theorem 1 or 2. In this section we discuss various extensions (or impossibility of them) of these polynomial solvable classes. Using NP-hardness results, we start by arguing in Sect. 5.1 that some of the assumptions of Theorem 1 or 2 can hardly be relaxed. We then discuss the extension of our results to minimization problems (Sect. 5.2), maximization of product of two functions (Sect. 5.3) and constrained problems (Sect. 5.4).

5.1 Limitations

In this section we show that unless $NP = P$ we generally cannot hope to solve problem (CFP) in polynomial time if we modify one assumption while keeping the other assumptions unchanged.

- It is not possible to replace the assumption "v is supermodular" by "v is submodular", while keeping all others assumptions unchanged: indeed by choosing $\varphi(t) = t$ and $\rho(t) = 1$, the problem (CFP) would become equivalent to maximizing a submodular function, which is known to be NP-hard. A similar restriction holds for the assumption "w is submodular".
- By the equivalence of the assumptions "φ is increasing and v is supermodular" and "φ is decreasing and v is submodular" (see Sect. 1), it follows that it is not possible to replace the assumption "φ is increasing" by "φ is decreasing". A similar statement could be made for the function ρ.
- It is not possible to remove the assumption that $\rho(t) > 0$ for all t. This follows from the following result of Hansen et al. [31]:

Proposition 17. *The problem*

$$\max_{x \in B_n} \quad \frac{a_0 + \sum_{j=1}^{n} a_j x_j}{b_0 + \sum_{j=1}^{n} b_j x_j}$$

is NP-hard unless the denominator is of the same sign for all $x \in B_n$.

(If the denominator is of the same sign for all $x \in B_n$, the above problem can be solved in linear time as shown by Hansen et al. [31]).

- The following observation involves two assumptions: it is not possible to replace the assumption "φ is convex" by the assumption "φ is concave", while at the same time removing the assumption that φ is increasing. To show this, we need to introduce the following well-known NP-hard problem SUBSET SUM (Garey and Johnson [23]):

  Input: n positive integers $s_1, s_2, \ldots, s_n$; an integer S.
  Question: does there exist a subset I of the index set $\{1, 2, \ldots, n\}$ such that $\sum_{i \in I} s_i = S$?

  We define $\varphi(t) = -|t|$, $\rho(t) = 1$ and $v(x) = \sum_{i \in I} s_i x_i - S$. Clearly the answer to SUBSET-SUM is yes if and only if the maximum of problem (CFP) is 0. The function φ is concave for all t and is increasing for $t < 0$ and decreasing for $t \geq 0$. A similar observation can be made for function ρ.

5.2 Minimization Problems

Consider the minimization version of problem (CFP):

$$\text{(CFPmin)} \qquad \min_{x \in B_n} \quad \frac{(\varphi \circ v)(x)}{(\rho \circ w)(x)}.$$

If $\varphi \circ v$ is strictly positive on B_n, we can use the equivalence

$$\min_{x \in B_n} \quad \frac{(\varphi \circ v)(x)}{(\rho \circ w)(x)} \qquad \Leftrightarrow \qquad \max_{x \in B_n} \quad \frac{(\rho \circ w)(x)}{(\varphi \circ v)(x)} \tag{32}$$

to derive sufficient conditions on (φ, ρ, v, w) for polynomial solvability of problem (CFPmin) from Theorem 1. However if $\varphi \circ v$ can take positive and negative values on B_n, equivalence (32) is not anymore true.

A similar analysis to the one done for the maximization problem results in the polynomial solvable classes described by Fig. 3. Figure 3 must be read as follows:

1. $(\rho \circ w)(x) > 0$ for $x \in B_n$;	
2. it is possible to evaluate the objective function in polynomial time;	
3. v and w take integral values on B_n;	
4. v and $-w$ are submodular;	
5. one of the following conditions is satisfied:	
• v or w takes a polynomial number of distinct values on B_n;	
• v and w are both linear;	
• v or w is monotone and the application $x \mapsto (v(x), w(x))$ is weakly bijective;	
6. $\min_{x \in B_n} (\varphi \circ v)(x) \leq 0$;	9. $(\varphi \circ v)(x) > 0$ for all $x \in B_n$;
7. φ and $-\rho$ are increasing;	10. φ and ρ are increasing;
8. $-\varphi$ and $-\rho$ are convex;	11. $-\varphi$ and ρ are convex.

Fig. 3 Description of the polynomial solvable classes for the minimization problem

1. $(\varphi_1 \circ v_1)(x) > 0$ for $x \in B_n$;	
2. it is possible to evaluate the objective function in polynomial time;	
3. v_1 and v_2 take integral values on B_n;	
4. v_1 and v_2 are supermodular;	
5. one of the following conditions is satisfied:	
• v_1 or v_2 takes a polynomial number of distinct values on B_n;	
• v_1 and v_2 are both linear;	
• v_1 or v_2 is monotone and the application $x \mapsto (v_1(x), v_2(x))$ is weakly bijective;	
6. $\max_{x \in B_n} (\varphi_2 \circ v_2)(x) \geq 0$;	9. $(\varphi_2 \circ v_2)(x) < 0$ for all $x \in B_n$;
7. φ_1 and φ_2 are increasing;	10. φ_1 and $-\varphi_2$ are increasing;
8. φ_1 and φ_2 are convex;	11. φ_1 and $-\varphi_2$ are convex.

Fig. 4 Description of the polynomial solvable classes for the problem of product maximization

if an instance of problem (CFPmin) satisfies the conditions 1–8 or satisfies the conditions 1–5 and 9–11, then the instance can be solved in polynomial time.

5.3 *Maximization of the Product of Two Composed Functions*

Consider the function $\sigma(t) = \frac{1}{\rho(-t)}$: if ρ is increasing and concave, function σ is increasing and convex. Problem (CFP) can then be reformulated as the maximization of the product of two functions:

$$\max_{x \in B_n} \quad \Big((\varphi_1 \circ v_1)(x)\Big)\Big((\varphi_2 \circ v_2)(x)\Big). \tag{33}$$

Figure 4 expresses the conditions of Theorems 1 and 2 in this new setting. Note that since (φ_1, v_1) and (φ_2, v_2) play a symmetrical role in (33), the first assumption in Fig. 4 could be replaced by $(\varphi_2 \circ v_2)(x) > 0$ for $x \in B_n$. However assumptions 6, 9, 10 and 11 should be modified accordingly.

We mention that the case where φ_1 and φ_2 are the identity functions and v_1 and v_2 are linear functions was studied by Hammer et al. [30].

5.4 Constrained Problems

In this last subsection we consider the constrained problem obtained from (CFP) by replacing the set B_n by a strict subset $T \subset B_n$. It can be verified that the sufficient condition of Proposition 12 remains valid. As soon as submodularity is involved, however, we usually need that T is a *sublattice* of B_n. A *sublattice* of B_n is a set T such that the following implication holds

$$x, y \in T \Rightarrow x \vee y \in T \text{and} x \wedge y \in T.$$

Most of the algorithms for supermodular maximization can be modified to optimize over a sublattice T without increase in the complexity, see McCormick [39]. When T is not a sublattice but can be expressed as the union of a polynomial number of sublattices, then it is possible to solve the constrained problem in polynomial time by running a supermodular maximization algorithm on each sublattice of the union and take the best answer. This is, for example, the case when $T = B_n \setminus \{(0), (1)\}$. In that case maximizing over T can be done via $O(n)$ calls to a SFM algorithm. See again McCormick [39] and also Goemans and Ramakrishnan [24].

When $T = T_{\rho_1,\rho_2}$ with $T_{\rho_1,\rho_2} = \{x \in B_n : \rho_1 \leq \sum_{i=1}^{n} x_i \leq n - \rho_2\}$ where ρ_1 and ρ_2 are fixed integers, it is possible to solve the constrained version of problem (CFP) in polynomial time by reducing it to $\binom{n}{\rho_1}\binom{n}{\rho_2}$ unconstrained submodular maximization problems with $n - \rho_1 - \rho_2$ variables, obtained by considering all possible ways to fix ρ_1 variables to 1 and ρ_2 variables to 0. We gave an illustration of this technique for $\rho_1 = 2$ and $\rho_2 = 0$ in Sect. 4.3. Note that B_n is equal to $T_{0,0}$.

6 Conclusion

We have presented a class of 0–1 fractional programming problems that are solvable in polynomial time. A nice particularity of the algorithm is that the candidate solution set is defined by only two of the four functions defining the objective function, which allow for low complexity if these two (supermodular) functions have a low degree representation (linear, quadratic, etc.). On the other hand, the two other functions may be more complicated, possibly non-rational, provided that their value can be evaluated and compared in polynomial time.

A lot of work remains to be done. On the practical side, it would be of course interesting to find real application problems where this approach can yield a polynomial algorithm. The additive clustering problem used to illustrate this method is a potential candidate but much work is needed to pass from 1 cluster (as illustrated in this paper) to m clusters. More generally, the question of the complexity of this problem for fixed m or when m is part of the input remains open.

On the theoretical side, several questions seem to worth of further study.

- The simplest problem (CFP) that is not fully understood occurs when φ and ρ are the identity functions, v is a quadratic supermodular function and w is a linear function, strictly positive on B_n. If neither v nor w is monotone, and each function takes more than a polynomial number of distinct values on B_n, there is no guarantee that the function Φ will have a polynomial number of breakpoints. Is it possible to either prove that the number of breakpoints will always be polynomial or to construct an example with a super-polynomial number of breakpoints? Carstensen [8], building on a result from Zadeh [56], proves that for any n there exists a parametric minimum cut problem on a graph G_n with $2n + 2$ nodes and $n^2 + n + 2$ arcs that has an exponential number of breakpoints. Unfortunately this network has some arcs with negative capacity, and thus does not seem to be usable to answer the above question.
- Except when one of the functions v or w takes a polynomial number of distinct values, the size of the set X^+ is always $O(n)$. This is mostly related to what we called the Monotone Optimal Solution Property. Does there exist problems where the size is larger, for example, $O(n \log n)$ or $O(n^2)$?
- We have shown in Sect. 5.3 that problem (CFP) can be reformulated as the maximization of the product of two functions of supermodular functions. Can we identify nontrivial polynomially solvable classes of the maximization of the product of p functions of supermodular functions, with $p > 2$?
- Can we find a relation between Dinkelbach's algorithm applied to problem (CFP) and Dinkelbach's algorithm applied to problem $\max_{x \in B_n} \frac{v(x)}{w(x)}$? More precisely, given a guess $\lambda_0 = \frac{(\varphi \circ v)(x^0)}{(\rho \circ w)(x^0)}$ for the optimal value of problem (CFP), can we deduce a λ_0' such that solving problem PARAM(λ_0') will yield an optimal solution $x^1 \in B_n$ that is guaranteed to satisfy $\frac{(\varphi \circ v)(x^1)}{(\rho \circ w)(x^1)} > \lambda_0$ when some termination criteria is not satisfied? One of the difficulties in attacking this question is that the functions v and w can be defined up to an additive constant, resulting in an infinite family of problems PARAM(λ).

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs (1993)
2. Arabie, P., Carroll, J.D.: MAPCLUS: a mathematical programming approach to fitting the ADCLUS model. Psychometrika **45**(2), 211–235 (1980)
3. Balinski, M.L.: On a selection problem. Manag. Sci. **17**, 230–231 (1970)
4. Berge, J.M.F.T., Kiers, H.A.L.: A comparison of two methods for fitting the INDCLUS model. J. Classif. **22**(2), 273–286 (2005)
5. Billionnet, A., Minoux, M.: Maximizing a supermodular pseudoboolean function: a polynomial algorithm for supermodular cubic functions. Discrete Appl. Math. **12**, 1–11 (1985)
6. Blömer, J.: Computing sums of radicals in polynomial time. In: 32nd Annual Symposium on Foundations of Computer Science, San Juan, PR, 1991, pp. 670–677. IEEE Computer Society Press, Los Alamitos (1991)

7. Carroll, J.D., Arabie, P.: INDCLUS: an individual differences generalization of the ADCLUS model and the MAPCLUS algorithm. Psychometrika **48**, 157–169 (1983)
8. Carstensen, P.J.: Complexity of some parametric integer and network programming problems. Math. Program. **26**(1), 64–75 (1983)
9. Chang, C.-T.: On the polynomial mixed 0-1 fractional programming problems. Eur. J. Oper. Res. **131**(1), 224–227 (2001)
10. Charnes, A., Cooper, W.W.: Programming with linear fractional functionals. Naval Res. Logist. Q. **9**, 181–186 (1962)
11. Chaturvedi, A., Carroll, J.D.: An alternating combinatorial optimization approach to fitting the INDCLUS and generalized INDCLUS models. J. Classif. **11**, 155–170 (1994)
12. Correa, J.R., Fernandes, C.G., Wakabayashi, Y.: Approximating a class of combinatorial problems with rational objective function. Math. Program. **124**(1–2, Ser. B), 255–269 (2010)
13. Craven, B.D.: Fractional Programming. Helderman Verlag, Berlin (1988)
14. Cunningham, W.H.: On submodular function minimization. Combinatorica **5**(3), 185–192 (1985)
15. Desarbo, W.S.: GENNCLUS: new models for general nonhierarchical clustering analysis. Psychometrika **47**(4), 449–475 (1982)
16. Dinkelbach, W.: On nonlinear fractional programming. Manag. Sci. **13**, 492–498 (1967)
17. Eisner, M.J., Severance, D.G.: Mathematical techniques for efficient record segmentation in large shared databases. J. Assoc. Comput. Mach. **23**(4), 619–635 (1976)
18. Feige, U., Mirrokni, V.S., Vondrák, J.: Maximizing non-monotone submodular functions. SIAM J. Comput. **40**(4), 1133–1153 (2011)
19. Frenk, J.B.G., Schaible, S.: Fractional programming. In: Handbook of Generalized Convexity and Generalized Monotonicity. Nonconvex Optimization and Its Applications, vol. 76, pp. 335–386. Springer, New York (2005)
20. Frenk, J.B.G., Schaible, S.: Fractional programming. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 1080–1091. Springer, Berlin (2009)
21. Gallo, G., Simeone, B.: On the supermodular knapsack problem. Math. Program. **45**(2, Ser. B), 295–309 (1989)
22. Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. SIAM J. Comput. **18**(1), 30–55 (1989)
23. Garey, M.R., Johnson, D.S.: Computers and Intractability, A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco (1979)
24. Goemans, M.X., Ramakrishnan, V.S.: Minimizing submodular functions over families of sets. Combinatorica **15**(4), 499–513 (1995)
25. Granot, F., McCormick, S.T., Queyranne, M., Tardella, F.: Structural and algorithmic properties for parametric minimum cuts. Math. Program. **135**(1–2, Ser. A), 337–367 (2012)
26. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2), 169–197 (1981)
27. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Algorithms and Combinatorics: Study and Research Texts, vol. 2. Springer, Berlin (1988)
28. Gusfield, D.M.: Sensitivity analysis for combinatorial optimization. Ph.D. thesis, University of California, Berkeley (1980)
29. Hammer, P.L., Rudeanu, S.: Boolean Methods in Operations Research and Related Areas. Springer, Berlin (1968)
30. Hammer, P.L., Hansen, P., Pardalos, P.M., Rader, D.J.: Maximizing the product of two linear functions in $0-1$ variables. Optimization **51**(3), 511–537 (2002)
31. Hansen, P., Poggi de Aragão, M.V., Ribeiro, C.C.: Hyperbolic 0-1 programming and query optimization in information retrieval. Math. Program. **52**(2, Ser. B), 255–263 (1991)
32. Hansen, P., Jaumard, B., Meyer, C.: Exact sequential algorithms for additive clustering. Technical Report G-2000-06, GERAD (March 2000)
33. Hochbaum, D.S.: Polynomial time algorithms for ratio regions and a variant of normalized cut. IEEE Trans. Pattern Anal. Mach. Intell. **32**(5), 889–898 (2010)

34. Ivănescu (Hammer), P.L.: Some network flow problems solved with pseudo-Boolean programming. Oper. Res. **13**, 388–399 (1965)
35. Iwata, S.: A faster scaling algorithm for minimizing submodular functions. SIAM J. Comput. **32**(4), 833–840 (2003)
36. Iwata, S., Fleischer, L., Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. ACM **48**(4), 761–777 (2001)
37. Kiers, H.A.L.: A modification of the SINDCLUS algorithm for fitting the ADCLUS and INDCLUS. J. Classif. **14**(2), 297–310 (1997)
38. Lee, M., Navarro, D.: Minimum description length and psychological clustering models. In: Grunwald, P., Myung, I., Pitt, M. (eds.) Advances in Minimum Description Length Theory and Applications. Neural Information Processing Series MIT Press, pp. 355–384 (2005). https://mitpress.mit.edu/books/advances-minimum-description-length
39. McCormick, S.T.: Chapter 7. Submodular function minimization. In: Aardal, K., Nemhauser, G.L., Weismantel, R. (eds.) Handbook on Discrete Optimization, pp. 321–391. Elsevier, Amsterdam (2005). Version 3a (2008). Available at http://people.commerce.ubc.ca/faculty/mccormick/sfmchap8a.pdf
40. Mirkin, B.G.: Additive clustering and qualitative factor analysis methods for similarity matrice. J. Classif. **4**, 7–31 (1987). Erratum, J. Classif. **6**, 271–272 (1989)
41. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)
42. Orlin, J.B.: A faster strongly polynomial time algorithm for submodular function minimization. Math. Program. **118**(2, Ser. A), 237–251 (2009)
43. Picard, J.C., Queyranne, M.: A network flow solution to some nonlinear $0-1$ programming problems, with applications to graph theory. Networks **12**, 141–159 (1982)
44. Qian, J., Wang, C.A.: How much precision is needed to compare two sums of square roots of integers? Inf. Process. Lett. **100**(5), 194–198 (2006)
45. Radzik, T.: Fractional combinatorial optimization. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 1077–1080. Springer, Berlin (2009)
46. Rhys, J.M.W.: A selection problem of shared fixed costs and network flows. Manag. Sci. **17**, 200–207 (1970)
47. Schaible, S.: Analyse und Anwendungen von Quotientenprogrammen, ein Beitrag zur Planung mit Hilfe der nichtlinearen Programmierung. Mathematical Systems in Economics, vol. 42. Verlag Anton Hain, Königstein/Ts. (1978)
48. Schaible, S., Shi, J.: Recent developments in fractional programming: single-ratio and max-min case. In: Nonlinear Analysis and Convex Analysis, pp. 493–506. Yokohama Publishers, Yokohama (2004)
49. Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Comb. Theory Ser. B **80**(2), 346–355 (2000)
50. Shepard, R.N., Arabie, P.: Additive clustering: representation of similarities as combinations of discrete overlapping properties. Psychol. Rev. **86**(2), 87–123 (1979)
51. Stancu-Minasian, I.M.: Fractional Programming: Theory, Methods, and Applications. Kluwer, Dordrecht (1997)
52. Stancu-Minasian, I.M.: A sixth bibliography of fractional programming. Optimization **55**(4), 405–428 (2006)
53. Topkis, D.M.: Minimizing a submodular function on a lattice. Oper. Res. **26**(2), 305–321 (1978)
54. Topkis, D.M.: Supermodularity and Complementarity. Frontiers of Economic Research. Princeton University Press, Princeton (1998)
55. Ursulenko, O.: Exact methods in fractional combinatorial optimization. ProQuest LLC, Ann Arbor, MI. Ph.D. thesis, Texas A&M University (2009)
56. Zadeh, N.: A bad network problem for the simplex method and other minimum cost flow algorithms. Math. Program. **5**, 255–266 (1973)

Experiments with a Non-convex Variance-Based Clustering Criterion

Rodrigo F. Toso, Evgeny V. Bauman, Casimir A. Kulikowski, and Ilya B. Muchnik

Abstract This paper investigates the effectiveness of a variance-based clustering criterion whose construct is similar to the popular minimum sum-of-squares or k-means criterion, except for two distinguishing characteristics: its ability to discriminate clusters by means of quadratic boundaries and its functional form, for which convexity does not hold. Using a recently proposed iterative local search heuristic that is suitable for general variance-based criteria—convex or not, the first to our knowledge that offers such broad support—the alternative criterion has performed remarkably well. In our experimental results, it is shown to be better suited for the majority of the heterogeneous real-world data sets selected. In conclusion, we offer strong reasons to believe that this criterion can be used by practitioners as an alternative to k-means clustering.

Keywords Clustering • Variance-based discriminants • Iterative local search

1 Introduction

Given a data set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of d-dimensional unlabeled samples, the clustering problem seeks a partition of $\mathcal{D}$ into k nonempty clusters such that the most similar samples are aggregated into a common cluster. We follow the

R.F. Toso (✉) • C.A. Kulikowski
Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA
e-mail: rtoso@cs.rutgers.edu; kulikows@cs.rutgers.edu

E.V. Bauman
Markov Processes International, Summit, NJ 07901, USA
e-mail: evbauman@markovprocesses.com

I.B. Muchnik
DIMACS, Rutgers University, Piscataway, NJ 08854, USA
e-mail: muchnik@dimacs.rutgers.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_3,

variational approach to clustering, where the quality of a clustering is evaluated by a criterion function (or functional), and the optimization process consists in finding a k-partition that minimizes such functional [2]. Perhaps, the most successful criteria for this approach are based on the sufficient statistics of each cluster $\mathcal{D}_i$, that is, their sample prior probabilities $\hat{p}_{\mathcal{D}_i}$, means $\hat{\boldsymbol{\mu}}_{\mathcal{D}_i}$, and variances $\hat{\sigma}^2_{\mathcal{D}_i}$, which yield not only mathematically motivated but also perceptually confirmable descriptions of the data. In general, such criteria are derived from the equation

$$J(\hat{p}_{\mathcal{D}_1}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_1}, \hat{\sigma}^2_{\mathcal{D}_1}, \ldots, \hat{p}_{\mathcal{D}_k}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_k}, \hat{\sigma}^2_{\mathcal{D}_k}) = \sum_{i=1}^{k} J_{\mathcal{D}_i}(\hat{p}_{\mathcal{D}_i}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}, \hat{\sigma}^2_{\mathcal{D}_i}). \tag{1}$$

In this paper, we focus on functional $J_3 = \sum_{i=1}^{k} \hat{p}^2_{\mathcal{D}_i} \hat{\sigma}^2_{\mathcal{D}_i}$, proposed by Kiseleva et al. [12]. Other examples include the minimum sum-of-squares criterion $J_1 = \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{D}_i} ||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2 = \sum_{i=1}^{k} \hat{p}_{\mathcal{D}_i} \hat{\sigma}^2_{\mathcal{D}_i}$ and Neyman's [17] variant $J_2 = \sum_{i=1}^{k} \hat{p}_{\mathcal{D}_i} \hat{\sigma}_{\mathcal{D}_i}$ that is rooted in the theory of sampling. The key difference from the traditional J_1 to J_2 and J_3 lies in the decision boundaries that discriminate the clusters: those in the latter are quadratic—the intra-cluster variances (dispersions), derived from the second normalized cluster moments, are also taken into account upon discrimination—and therefore more flexible than the linear discriminants employed by the former, which only takes into account the means (centers) of the clusters.

The distinctive feature of criterion J_3 is the lack of convexity, given that both J_1 and J_2 are convex [15]. This claim has a deep impact in the practical application of the criterion, given that, to our knowledge, no simple clustering heuristic offers support for non-convex functionals, including the two-phase "k-means" clustering algorithm of Lloyd [14]. (It is worth mentioning that the two-phase algorithm was extended so as to optimize J_2, thus enabling an initial computational study of the criterion [15].) Even though there exist more complex algorithms that provide performance guarantees for J_3 (see, e.g., [7, 21]), we could not find in the literature any experimental study validating them in practice, perhaps due to their inherent complexities.

There was, to our knowledge, a gap of more than 25 years between the introduction of the criterion J_3 in 1986 [12] and the first implementation of an algorithm capable of optimizing it, published in 2012 by Toso et al. [23]. Their work has introduced a generalized version of the efficient iterative minimum sum-of-squares local search heuristic studied by Späth [22] in 1980 which, in turn, is a variant of the online one-by-one procedure of MacQueen [16]. (An online algorithm does not require the whole data set as input—it reads the input sample by sample.)

This paper serves two purposes: first and foremost, it provides evidence of the effectiveness of the criterion J_3 when contrasted with J_1 and J_2 on real-world data sets; additionally, it offers an overview of the local search heuristic for general variance-based criterion minimization first published in [23]. Since the minimum sum-of-squares criterion is widely adopted by practitioners—in our view because

the k-means algorithm is both effective and easy to implement—we offer strong reasons to believe that J_3 can be successfully employed in real-world clustering tasks, given that it also shares these very same features.

Our paper is organized as follows. In the next section, we establish the notation and discuss the background of our work. The experimental evaluation appears in Sect. 3, and conclusions are drawn in Sect. 4.

2 Variance-Based Clustering: Criteria and an Optimization Heuristic

We begin this section by establishing the notation adopted throughout our work. Next, we review the framework of variance-based clustering and describe the three main criteria derived from it, including functional J_3. We then conclude with a brief discussion about the local search heuristic that can optimize any criterion represented by Eq. (1).

2.1 Notation

Given an initial k-partition of $\mathcal{D}$, let the number of samples in a given cluster $\mathcal{D}_i$ be $n_{\mathcal{D}_i}$; this way, the prior probability of $\mathcal{D}_i$ is estimated as $\hat{p}_{\mathcal{D}_i} = \frac{n_{\mathcal{D}_i}}{n}$. The first and second sample central moments of the clusters are given by

$$\mathcal{M}_{\mathcal{D}_i}^{(1)} = \hat{\boldsymbol{\mu}}_{\mathcal{D}_i} = \frac{1}{n_{\mathcal{D}_i}} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}, \text{ and} \tag{2}$$

$$\mathcal{M}_{\mathcal{D}_i}^{(2)} = \frac{1}{n_{\mathcal{D}_i}} \sum_{\mathbf{x} \in \mathcal{D}_i} ||\mathbf{x}||^2, \tag{3}$$

respectively, where the former is the sample mean of the cluster $\mathcal{D}_i$. It follows that the sample variance of $\mathcal{D}_i$ is computed by

$$\hat{\sigma}_{\mathcal{D}_i}^2 = \mathcal{M}_{\mathcal{D}_i}^{(2)} - ||\mathcal{M}_{\mathcal{D}_i}^{(1)}||^2. \tag{4}$$

Here, $\hat{p}_{\mathcal{D}_i} \in \mathbb{R}$, $\hat{\boldsymbol{\mu}}_{\mathcal{D}_i} \in \mathbb{R}^d$, and $\hat{\sigma}_{\mathcal{D}_i}^2 \in \mathbb{R}$, for all $i = 1, \ldots, k$.

2.2 Variance-Based Clustering Criteria

Among the criterion functions derived from Eq. (1) is the minimum sum-of-squares clustering (MSSC) criterion, since

$$\begin{aligned}\min \text{MSSC} &= \min \sum_{i=1}^{k} \sum_{\mathbf{x}\in\mathcal{D}_i} ||x - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2 \qquad (5)\\ &= \min \frac{1}{n} \sum_{i=1}^{k} \frac{n_{\mathcal{D}_i}}{n_{\mathcal{D}i}} \sum_{\mathbf{x}\in\mathcal{D}_i} ||x - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2\\ &= \min \sum_{i=1}^{k} \frac{n_{\mathcal{D}_i}}{n} \frac{1}{n_{\mathcal{D}i}} \sum_{\mathbf{x}\in\mathcal{D}_i} ||x - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2\\ &= \min \sum_{i=1}^{k} \hat{p}_{\mathcal{D}_i} \hat{\sigma}^2_{\mathcal{D}_i} = J_1 \qquad (6)\end{aligned}$$

The functional form in Eq. (5) is called *membership* or *discriminant function* since it explicitly denotes the similarity of a sample with respect to a cluster. On the other hand, Eq. (6) quantifies the similarity of each cluster directly. Note that the former is in fact the gradient of the latter, which, in this case, is convex [2]. In J_1, the separating hyperplane (also known as decision boundary) between two clusters $\mathcal{D}_i$ and $\mathcal{D}_j$ with respect to a sample $\mathbf{x}$ is given by the equation

$$||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2 - ||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_j}||^2 = 0. \qquad (7)$$

Let us now turn our attention to a criterion proposed by Neyman [17] for one-dimensional sampling:

$$J_2 = \sum_{i=1}^{k} \hat{p}_{\mathcal{D}_i} \sqrt{\hat{\sigma}^2_{\mathcal{D}_i}}. \qquad (8)$$

Recently, J_2 was generalized so as to support multidimensional data and proven to be convex [15]. The decision boundaries produced by criterion J_2 are given by Eq. (9) and, contrary to those of J_1, take into account the variance of the clusters for discrimination.

$$\left[\frac{\hat{\sigma}_{\mathcal{D}_i}}{2} + \frac{1}{2\hat{\sigma}_{\mathcal{D}_i}}(||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2)\right] - \left[\frac{\hat{\sigma}_{\mathcal{D}_j}}{2} + \frac{1}{2\hat{\sigma}_{\mathcal{D}_j}}(||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_j}||^2)\right] = 0. \qquad (9)$$

Finally, Kiseleva et al. [12] introduced the one-dimensional criterion function

$$J_3 = \sum_{i=1}^{k} \hat{p}_{\mathcal{D}_i}^2 \hat{\sigma}_{\mathcal{D}_i}^2, \tag{10}$$

whose decision boundaries are

$$\left[\hat{p}_{\mathcal{D}_i}\hat{\sigma}_{\mathcal{D}_i}^2 + \hat{p}_{\mathcal{D}_i}(||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_i}||^2)\right] - \left[\hat{p}_{\mathcal{D}_j}\hat{\sigma}_{\mathcal{D}_j}^2 + \hat{p}_{\mathcal{D}_j}(||\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{D}_j}||^2)\right] = 0. \tag{11}$$

The criterion was extended to multidimensional samples [21], but a recent manuscript has sketched a proof stating that convexity does not hold [15]. On a positive note, and similarly to J_2, the decision boundaries of J_3 make use of the variances of the clusters.

2.2.1 Non-convexity of J_3

The aforementioned proof relies on the following:

Theorem 1. *Assume that $f : \mathbb{R}^d \to \mathbb{R}$ is twice continuously differentiable. Then, f is convex if and only if its Hessian $\nabla^2 f(\mathbf{x})$ is positive semidefinite for all $\mathbf{x} \in \mathbb{R}^d$ [20].* □

It is enough to show that $J_3(\mathcal{D}_i) = \hat{p}_{\mathcal{D}_i}^2 \hat{\sigma}_{\mathcal{D}_i}^2 = M_{\mathcal{D}_i}^{(0)} M_{\mathcal{D}_i}^{(2)} - ||M_{\mathcal{D}_i}^{(1)}||^2$ is not convex, with $M_{\mathcal{D}_i}^{(m)}$ denoting the m-th non-normalized sample moment of cluster $\mathcal{D}_i$.

Theorem 2. *Functional J_3 is non-convex.*

Proof. The partial derivatives of $J_3(\mathcal{D}_i)$ are

$$\frac{\partial J_3(\mathcal{D}_i)}{\partial M_{\mathcal{D}_i}^{(0)}} = M_{\mathcal{D}_i}^{(2)},$$

$$\frac{\partial J_3(\mathcal{D}_i)}{\partial M_{\mathcal{D}_i}^{(1)}} = -2M_{\mathcal{D}_i}^{(1)}, \text{ and}$$

$$\frac{\partial J_3(\mathcal{D}_i)}{\partial M_{\mathcal{D}_i}^{(2)}} = M_{\mathcal{D}_i}^{(0)}.$$

Thus, Eq. (12) corresponds to the Hessian of the functional of a cluster.

$$\nabla^2 J_3(\mathcal{D}_i) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 0 \end{pmatrix} \tag{12}$$

We now proceed to show that the Hessian in Eq. (12) is not positive semidefinite, as required by Theorem 1.

Definition 1. A matrix $M \in \mathbb{R}^{d\times d}$ is called positive semidefinite if it is symmetric and $\mathbf{x}^\intercal M\,\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^d$.

We show that there exists an $\mathbf{x}$ such that

$$\mathbf{x}^T \begin{pmatrix} 0 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 0 \end{pmatrix} \mathbf{x} < 0,$$

contradicting Definition 1. Plugging $\mathbf{x} = (1\ 2\ 1)$ does the job, so $\nabla^2 J_3(\mathcal{D}_i)$ is not positive semidefinite and, by Theorem 1, J_3 is non-convex.

2.3 Algorithms

Let us begin with J_1. Although exact algorithms have been studied [8,9], minimizing Eq. (5) is NP-hard [5], and hence their use is restricted to small data sets. In general, except for approximation algorithms and local search heuristics with no performance guarantees, no other optimization technique shall be suitable to minimize Eq. (1) for large data sets unless $P = \text{NP}$.

The so-called k-means clustering algorithm is perhaps the most studied local search heuristic for the minimization of J_1 (disguised in Eq. (5)). In fact, k-means usually refers to (variants of) one of the following two algorithms. The first is an iterative two-phase procedure due to Lloyd [14] that is initialized with a k-partition and alternates two phases: (1) given the set of samples $\mathcal{D}$ and the k cluster centers, it reassigns each sample to the closest center; and (2) with the resulting updated k-partition, it updates the centers. This process is iteratively executed until a stopping condition is met. The second variant is an incremental one-by-one procedure which utilizes the first k samples of $\mathcal{D}$ as the cluster centers. Each subsequent sample is assigned to the closest center, which is then updated to reflect the change. This procedure was introduced by MacQueen [16] and is a single-pass, online procedure, where samples are inspected only once before being assigned to a cluster. In [22], an efficient iterative variant of this approach was given and can also be seen in Duda et al. [6] (cf. Chap. 9, Basic Iterative Minimum-Squared-Error Clustering). A comprehensive survey on the origins and variants of k-means clustering algorithms can be found in [3].

The main difference between the two approaches above is when the cluster centers are updated: in a separate phase, after all the samples have been considered (two-phase), or every time a sample is reassigned to a different cluster (one-by-one). It is here that the main drawbacks of the one-by-one method appear: the computational time required to update the cluster centers after each sample is

reassigned must increase in comparison with the two-phase approach, whereas the ability to escape from local minima has also been questioned [6]. Both issues have been addressed in [23], where the former is shown to be true while the latter is debunked in an experimental analysis. In contrast, significant improvements have been made in the two-phase algorithm when tied with the minimum-squared error criterion, such as tuning it to run faster [11, 18] or to be less susceptible to local minima [4, 13], and also making it more general [19].

With J_2, since convexity holds, the two-phase procedure could be successfully employed in a thorough evaluation reported in [15].

At last, J_3. Lloyd's [14] two-phase, gradient-based heuristic cannot be applied due to the non-convexity of the functional, but MacQueen's [16] online approach is a viable direction. A randomized sampling-based approximation algorithm was introduced by Schulman [21], relying on dimensionality reduction to make effective use of an exact algorithm whose running time grows exponentially with the dimensionality of the data. Also, a deterministic approximation algorithm appears in [7]. To our knowledge, no implementations validating these approaches have been reported in the literature. In conclusion, there was no experimental progress with J_3 prior to [23] and this can certainly be attributed to the lack of a computationally viable algorithmic alternative.

2.3.1 An Iterative Heuristic for Variance-Based Clustering Criteria

In this section, we revisit the local search heuristic appearing in [23] for criteria in the class of Eq. (1), including those three studied throughout this paper. The algorithm combines the key one-by-one, monotonically decreasing, approach of MacQueen [16] with the iterative design of Späth [22], extending the efficient way to maintain and update the sufficient statistics of the clusters also used in the latter.

We first present in Algorithm 1 an efficient procedure to update a given functional value to reflect the case where an arbitrary sample $\mathbf{x} \in \mathcal{D}_j$ is reassigned to cluster $\mathcal{D}_i$ $(i \neq j)$; we use the notation $J^{(\mathbf{x}\to\mathcal{D}_i)}$ to indicate that $\mathbf{x}$, currently in cluster $\mathcal{D}_j$, is about to be considered in cluster $\mathcal{D}_i$. Similarly to [6, 22], we maintain the unnormalized statistics of each cluster, namely $n_{\mathcal{D}_i}$, the number of samples assigned to cluster $\mathcal{D}_i$, $\mathbf{m}_{\mathcal{D}_i} = \sum_{\mathbf{x}\in\mathcal{D}_i} \mathbf{x}$, and $s^2_{\mathcal{D}_i} = \sum_{\mathbf{x}\in\mathcal{D}_i} ||\mathbf{x}||^2$. Such equations not only can be efficiently updated when a sample is moved from one cluster to another but also allow us to compute or update the criterion function quickly. Note that in [6,22], only $n_{\mathcal{D}_i}$ and $\mathbf{m}_{\mathcal{D}_i}$ need to be maintained since the algorithms are tied with J_1 as in Eq. (5) (the minimum sum-of-squares variant).

The main clustering heuristic is shown in Algorithm 2. The procedure is initialized with a k-partition that is used to compute the auxiliary and the sample statistics in lines 1 and 2, respectively. In the main loop (lines 5–17), every sample $\mathbf{x} \in \mathcal{D}$ is considered as follows: Algorithm 1 is used to assess the functional value when the current sample is tentatively moved to each cluster $\mathcal{D}_1, \ldots, \mathcal{D}_k$ (lines 6–8).

Algorithm 1 Computes $J^{(\mathbf{x}\rightarrow\mathcal{D}_i)}$

Input: sample $\mathbf{x} \in \mathcal{D}_j$, target cluster $\mathcal{D}_i$, current criterion value J^*, and cluster statistics: $n_{\mathcal{D}_j}$, $\mathbf{m}_{\mathcal{D}_j}$, $s^2_{\mathcal{D}_j}$, $\hat{p}_{\mathcal{D}_j}$, $\hat{\boldsymbol{\mu}}_{\mathcal{D}_j}$, $\hat{\sigma}^2_{\mathcal{D}_j}$, $n_{\mathcal{D}_i}$, $\mathbf{m}_{\mathcal{D}_i}$, $s^2_{\mathcal{D}_i}$, $\hat{p}_{\mathcal{D}_i}$, $\hat{\boldsymbol{\mu}}_{\mathcal{D}_i}$, and $\hat{\sigma}^2_{\mathcal{D}_i}$.

Let $n'_{\mathcal{D}_j} := n_{\mathcal{D}_j} - 1$ and $n'_{\mathcal{D}_i} := n_{\mathcal{D}_i} + 1$.
Let $\mathbf{m}'_{\mathcal{D}_j} := \mathbf{m}_{\mathcal{D}_j} - \mathbf{x}$ and $\mathbf{m}'_{\mathcal{D}_i} := \mathbf{m}_{\mathcal{D}_i} + \mathbf{x}$.
Let $(s^2_{\mathcal{D}_j})' := s^2_{\mathcal{D}_j} - ||\mathbf{x}||^2$ and $(s^2_{\mathcal{D}_i})' := s^2_{\mathcal{D}_i} + ||\mathbf{x}||^2$.
Let $\hat{p}'_{\mathcal{D}_j} := \frac{n'_{\mathcal{D}_j}}{n}$ and $\hat{p}'_{\mathcal{D}_i} := \frac{n'_{\mathcal{D}_i}}{n}$.
Let $\hat{\boldsymbol{\mu}}'_{\mathcal{D}_j} := \frac{1}{n'_{\mathcal{D}_j}}\mathbf{m}_{\mathcal{D}'_j}$ and $\hat{\boldsymbol{\mu}}'_{\mathcal{D}_i} := \frac{1}{n'_{\mathcal{D}_i}}\mathbf{m}'_{\mathcal{D}_i}$.
Let $(\hat{\sigma}^2_{\mathcal{D}_j})' := \frac{1}{n'_{\mathcal{D}_j}}(s^2_{\mathcal{D}_j})' - ||\hat{\boldsymbol{\mu}}'_{\mathcal{D}_j}||^2$ and $(\hat{\sigma}^2_{\mathcal{D}_i})' := \frac{1}{n'_{\mathcal{D}_i}}(s^2_{\mathcal{D}_i})' - ||\hat{\boldsymbol{\mu}}'_{\mathcal{D}_i}||^2$.
Compute $J^{(\mathbf{x}\rightarrow\mathcal{D}_i)}$ with the updated statistics for clusters $\mathcal{D}_i$ and $\mathcal{D}_j$.

Algorithm 2 Minimizes a clustering criterion function

Input: an initial k-partition.

Compute $n_{\mathcal{D}_i}$, $\mathbf{m}_{\mathcal{D}_i}$, and $s^2_{\mathcal{D}_i}$ $\forall\ i = 1, \ldots, k$.
Compute $\hat{p}_{\mathcal{D}_i}$, $\hat{\boldsymbol{\mu}}_{\mathcal{D}_i}$, and $\hat{\sigma}^2_{\mathcal{D}_i}$ $\forall\ i = 1, \ldots, k$.
Set $J^* := J(\hat{p}_{\mathcal{D}_1}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_1}, \hat{\sigma}^2_{\mathcal{D}_1}, \ldots, \hat{p}_{\mathcal{D}_k}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_k}, \hat{\sigma}^2_{\mathcal{D}_k})$.
while convergence criterion not reached **do**
  for all $\mathbf{x} \in \mathcal{D}$ **do**
    for all $i \mid h_{\mathcal{D}_i}(\mathbf{x}) = 0$ **do**
      Compute $J^{(\mathbf{x}\rightarrow\mathcal{D}_i)}$ via Algorithm 1.
    end for
    if $\exists\ i \mid J^{(\mathbf{x}\rightarrow\mathcal{D}_i)} < J^*$ **then**
      Let $min = i \mid \min_i J^{(\mathbf{x}\rightarrow\mathcal{D}_i)}$. (i.e., $\mathbf{x} \rightarrow \mathcal{D}_{min}$ mostly improves J^*.)
      Let $j = i \mid h_{\mathcal{D}_i}(\mathbf{x}) = 1$. (i.e., $\mathcal{D}_j$ is the current cluster of $\mathbf{x}$.)
      Set $h_{\mathcal{D}_{min}}(\mathbf{x}) := 1$ and $h_{\mathcal{D}_j}(\mathbf{x}) := 0$. (i.e., assign $\mathbf{x}$ to cluster $\mathcal{D}_{min}$.)
      Update: $n_{\mathcal{D}_{min}}, n_{\mathcal{D}_j}, \mathbf{m}_{\mathcal{D}_{min}}, \mathbf{m}_{\mathcal{D}_j}\ s^2_{\mathcal{D}_{min}}, s^2_{\mathcal{D}_j}$.
      Update: $\hat{p}_{\mathcal{D}_{min}}, \hat{p}_{\mathcal{D}_j}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_{min}}, \hat{\boldsymbol{\mu}}_{\mathcal{D}_j}, \hat{\sigma}^2_{\mathcal{D}_{min}}, \hat{\sigma}^2_{\mathcal{D}_j}$.
      Set $J^* := J^{(\mathbf{x}\rightarrow\mathcal{D}_{min})}$.
    end if
  end for
end while

If there exists a cluster $\mathcal{D}_{\text{min}}$ for which the objective function can be improved, the sample is reassigned to such cluster and all the statistics are updated. The algorithm stops when a convergence goal is reached.

With Algorithm 1 running in $\Theta(d)$, the running time to execute one iteration of Algorithm 2 is $\Theta(nkd)$, the same of an iteration of the simple two-phase procedure [6]. In practice, though, due to the constant terms hidden in the analysis, the latter is consistently faster than the former. Results for clustering quality have shown that the two approaches offer comparable clusterings [23].

The algorithm is quite flexible and can be extended with one or more regularization terms, such as to balance cluster sizes. Given that it is simple to implement and quick to run, the method can also be employed as a local search procedure with a global optimization overlay, such as genetic algorithms.

3 Experimental Results

This section offers an overview of the results obtained in [23]. Algorithm 2 was implemented in C++, compiled with g++ version 4.1.2, and run on a single 2.3 GHz CPU with 128 GBytes of RAM. The algorithm was stopped when no sample was moved to a different cluster in a complete iteration.

A visualization-friendly two-dimensional instance illustrating the capabilities of J_3 is depicted in Fig. 1, offering a glimpse of how the decision boundaries of the three criteria in study discriminate samples. Clearly, J_2 and J_3 built quadratic boundaries around the central cluster (of smaller variance) and linear hyperplanes between the external clusters (of the same variance), since Eqs. (9) and (11) become linear when $\hat{\sigma}^2_{\mathcal{D}_i} = \hat{\sigma}^2_{\mathcal{D}_j}$. For J_1, all boundaries are linear and thus unable to provide a proper discrimination for the central cluster.

In Fig. 2 we plot the value of J_3 after each sample is inspected by the algorithm over the course of nine iterations on a synthetic data set, when the algorithm halted. The figure not only shows that the criterion is consistently reduced up until the second iteration but also provides a hint for a possible enhancement, suggesting that the algorithm could be stopped at the end of the third iteration. Since our focus is on J_3 itself, no running-time enhancements were made on the algorithm though.

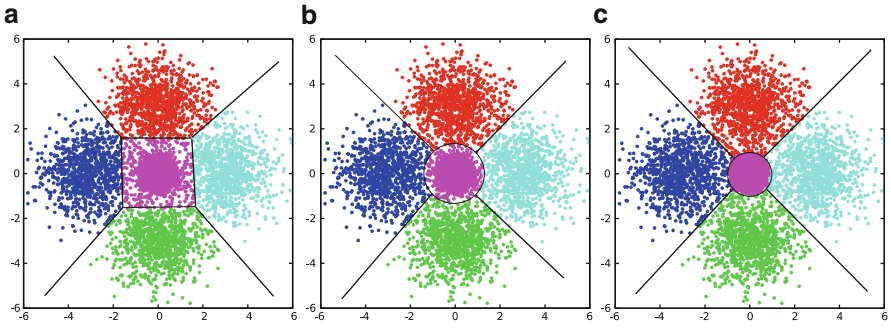


Fig. 1 Decision boundaries for a mixture of five equiprobable Gaussian distributions. The central cluster has a quarter of the variance of the external clusters. (**a**) Criterion J_1. (**b**) Criterion J_2. (**c**) Criterion J_3

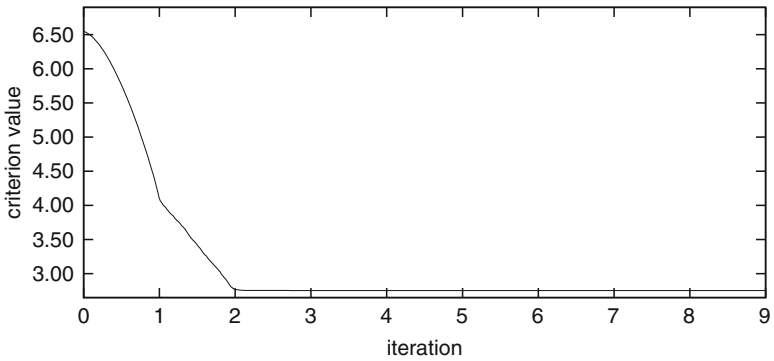


Fig. 2 Evolution of the criterion value for a randomly generated mixture of Gaussians with $n = 10{,}000$, $k = d = 50$, and $\sigma^2 = 0.05$. Each of the $k = 50$ centers was randomly placed inside a d-dimensional unit hypercube

3.1 Clustering Quality Analysis

In the subsequent experiments, we selected 12 real-world *classification* data sets (those with available class labels) from the UCI Machine Learning Repository [1] having fairly heterogeneous parameters as shown in Table 1. The available class labels from the selected data sets were used in order to compare the functionals under the following measures of clustering quality: accuracy, widely adopted by the classification community, and the Adjusted Rand Index or ARI [10], a pair-counting measure adjusted for chance that is extensively adopted by the clustering community. (See Vinh et al. [24].)

From the qualitative results in Table 1, we note that J_3 significantly outperforms both J_1 and J_2 on average, being about 2 % better than its counterparts in both quality measures. Although we chose not to display the individual standard deviations for each data set, the average standard deviation in accuracy across all data sets was 0.0294, 0.0291, and 0.0276 for J_1, J_2, and J_3, respectively; for ARI, 0.0284, 0.0282, and 0.0208, respectively. In this regard, J_3 also offered a more stable operation across the different initial solutions.

4 Summary and Future Research

This paper has shown promising qualitative results for a non-convex criterion function for clustering problems, obtaining outstanding results on heterogeneous data sets of various real-world applications including digit recognition, image segmentation, and discovery of medical conditions. We strongly believe that this criterion can be an excellent addition to applications involving exploratory data

Table 1 Description and solution quality for real-world data sets obtained from the UCI Repository [1]

	Parameters			Accuracy			Adjusted Rand Index		
Data set	k	d	n	J_1	J_2	J_3	J_1	J_2	J_3
Arcene	2	10000	200	0.6191	0.6173	**0.6750**	0.0559	0.0536	**0.1180**
Breast-cancer	2	30	569	0.8541	0.8735	**0.8770**	0.4914	0.5502	**0.5613**
Credit	2	42	653	0.5513	**0.5865**	0.5819	0.0019	**0.0226**	0.0193
Inflammations	4	6	20	0.6773	0.6606	**0.7776**	0.4204	0.4008	**0.6414**
Internet-ads	2	1558	2359	**0.8953**	0.8279	0.7961	**0.4975**	0.3434	0.2771
Iris	3	4	150	**0.8933**	**0.8933**	**0.8933**	**0.7302**	**0.7302**	0.7282
Lenses	2	6	24	**0.6036**	0.6011	0.6012	0.0346	0.0326	**0.0382**
Optdigits	10	64	5619	0.7792	0.7702	**0.7959**	0.6619	0.6498	**0.6810**
Pendigits	10	16	10992	0.6857	0.6960	**0.7704**	0.5487	0.5746	**0.6155**
Segmentation	7	19	2310	0.5612	0.5516	**0.5685**	0.3771	0.3758	**0.4028**
Spambase	2	57	4601	0.6359	**0.6590**	0.6564	0.0394	**0.0773**	0.0726
Voting	2	16	232	**0.8966**	0.8875	0.8865	**0.6274**	0.5988	0.5959
Average				0.7211	0.7187	**0.7400**	0.3739	0.3675	**0.3959**
Wins				4	3	7	3	3	7

Quality measures are averaged over 1,000 runs with random initial k-partitions

analysis, given its ability to discriminate clusters with quadratic boundaries based on their variance.

Future research paths include a more extensive experimentation with functionals J_2 and J_3 while also contrasting them with J_1 to better understand their strengths and weaknesses.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2009) http://archive.ics.uci.edu/ml/
2. Bauman, E.V., Dorofeyuk, A.A.: Variational approach to the problem of automatic classification for a class of additive functionals. Autom. Remote Control **8**, 133–141 (1978)
3. Bock, H.-H.: Origins and extensions of the k-means algorithm in cluster analysis. Electron. J. Hist. Probab. Stat. **4**(2) (2008)
4. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: Proceedings of the 15th International Conference on Machine Learning, pp. 91–99. Morgan Kaufmann Publishers, San Francisco (1998)
5. Brucker, P.: On the complexity of clustering problems. In: Optimization and Operations Research. Lecture Notes in Economics and Mathematical Systems, vol. 157, pp. 45–54. Springer, Berlin (1978)
6. Duda, R.O., Hart, P.E., Storck, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2000)
7. Efros, M., Schulman, L.J.: Deterministic clustering with data nets. Technical Report 04-050, Electronic Colloquium on Computational Complexity (2004)
8. Grotschel, M., Wakabayashi, Y.: A cutting plane algorithm for a clustering problem. Math. Program. **45**, 59–96 (1989)
9. Hansen, P., Jaumard, B.: Cluster analysis and mathematical programming. Math. Program. **79**, 191–215 (1997)
10. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)

11. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 881–892 (2002)
12. Kiseleva, N.E., Muchnik, I.B., Novikov, S.G.: Stratified samples in the problem of representative types. Autom. Remote Control **47**, 684–693 (1986)
13. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means algorithm. Pattern Recognit. **36**, 451–461 (2003)
14. Lloyd, S.P.: Least squares quantization in PCM. Technical report, Bell Telephone Labs Memorandum (1957)
15. Lytkin, N.I., Kulikowski, C.A., Muchnik, I.B.: Variance-based criteria for clustering and their application to the analysis of management styles of mutual funds based on time series of daily returns. Technical Report 2008-01, DIMACS (2008)
16. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
17. Neyman, J.: On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. J. R. Stat. Soc. **97**, 558–625 (1934)
18. Pelleg, D., Moore, A.: Accelerating exact k-means algorithms with geometric reasoning. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 277–281. ACM, New York (1999)
19. Pelleg, D., Moore, A.: x-means: extending k-means with efficient estimation of the number of clusters. In: Proceedings of the 17th International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann Publishers, San Francisco (2000)
20. Ruszczyński, A.: Nonlinear Programming. Princeton University Press, Princeton (2006)
21. Schulman, L.J.: Clustering for edge-cost minimization. In: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, pp. 547–555. ACM, New York (2000)
22. Späth, H.: Cluster Analysis Algorithms for Data Reduction and Classification of Objects. E. Horwood, Chichester (1980)
23. Toso, R.F., Kulikowski, C.A., Muchnik, I.B.: A heuristic for non-convex variance-based clustering criteria. In: Klasing, R. (ed.) Experimental Algorithms. Lecture Notes in Computer Science, vol. 7276, pp. 381–392. Springer, Berlin (2012)
24. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1073–1080. ACM, New York (2009)

Strategy-Proof Location Functions on Finite Graphs

F.R. McMorris, Henry Martyn Mulder, and Fred S. Roberts

This paper is dedicated to our friend and colleague Boris Mirkin on the occasion of his 70th birthday

Abstract A location function on a finite graph takes a set of most preferred locations (vertices of the graph) for a set of users, and returns a set of locations satisfying conditions meant to please the entire user set as much as possible. A strategy-proof location function is one for which it never benefits a user to report a suboptimal preferred location. We introduce four versions of strategy-proof and prove some preliminary results focusing on two well-known location functions, the median and the center.

Keywords Location function • Center • Median • Strategy-proof

F.R. McMorris (✉)
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA

Department of Mathematics, University of Louisville, Louisville, KY 40292, USA
e-mail: mcmorris@iit.edu

H.M. Mulder
Econometrisch Instituut, Erasmus Universiteit, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
e-mail: hmmulder@few.eur.nl

F.S. Roberts
DIMACS Center, Rutgers University, Piscataway, NJ 08854, USA
e-mail: froberts@dimacs.rutgers.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_4,

1 Introduction

A common problem for many location studies is to find a location or set of locations that satisfies a group of customers in a way that is as good as possible, usually by maximizing or minimizing various optimization criteria. The customers are often viewed as "voters" where each one reports a preferred location on a graph, and the location function returns a set of "winners." Most of the work done in this area focuses on developing algorithms to find these optimal location vertices, but in recent years, there have been axiomatic studies of the procedures themselves. This is the approach we take in this note. We seek to understand those location functions that encourage voters/customers to report their true location preferences. That is, no voter j should be able to improve the outcome (from j's point-of-view) by reporting a suboptimal location in their vote. Standard terminology labels these functions as being "strategy-proof," and the literature on this topic is extensive. For example see [13] for the many references therein. Our goal is to develop the notion of strategy-proofness as it pertains to the vertex set of a finite graph with added graph-theoretic structure. We deviate from many studies (e.g. see [1, 12]) by requiring all locations and customers to be on vertices of the graph, and that the edges have no real-valued lengths assigned to them. We introduce four precise concepts of strategy-proofness in our context and give some preliminary results about them. Specifically, we illustrate the concepts by looking at two well-known location functions, the median and the center, and we study these functions on several classes of graphs.

2 Preliminaries and an Elementary Result

Throughout we let $G = (V, E)$ be a finite, connected graph without loops or multiple edges, with vertex set V and edge set E. The *distance* $d(u, v)$ between two vertices u and v of G is the length of a shortest u, v-path, so that (V, d) is a finite metric space. If $X \subseteq V$ and $v \in V$, then we set $d(v, X) = \min\{d(v, x) : x \in X\}$. Let k be a positive integer. Sequences in V^k are called *profiles* and a generic one is denoted $\pi = (x_1, \ldots, x_k)$. Let $\{\pi\}$ be the set of distinct vertices appearing in π and $|\pi|$ be number of elements in $\{\pi\}$. By $\pi[x_j \to w]$ we denote the profile obtained from $\pi = (x_1, \ldots, x_j, \ldots, x_k)$ by replacing x_j by w. So $\pi[x_j \to w] = (x_1, \ldots, x_{j-1}, w, x_{j+1}, \ldots, x_k)$, for $1 < j < k$, and $\pi[x_1 \to w] = (w, x_2, \ldots, x_k)$, and $\pi[x_k \to w] = (x_1, \ldots, x_{k-1}, w)$.

Without any conditions imposed, a *location function* (*of order k*) on G is simply a mapping $L_V : V^k \to 2^V \backslash \{\emptyset\}$, where 2^V denotes the set of all subsets of V. When the set V is clear from the context, we will write L instead of L_V. A *single-valued location function* on G is a function of the form $L : V^k \to V$. (Notice that a single-valued L can be viewed as requiring $|L(\pi)| = 1$ for all π.) Given a profile π, we can think of x_i as the reported location desired by customer (or voter) i, and $L(\pi)$

as the set of locations produced by the function L. To measure how "close" a vertex x is to a given profile $\pi = (x_1, \ldots, x_k)$, the values of $s(x, \pi) = \sum_{i=1}^{k} d(x, x_i)$ and $e(x, \pi) = \max\{d(x, x_1), \ldots, d(x, x_k)\}$ have been used often. We will be concerned with two well-studied location functions (e.g., see [4–6]) which return vertices close, in the previous sense, to a given profile. The *center function* is the location function Cen : $V^k \to 2^V \backslash \{\emptyset\}$ defined by $\text{Cen}(\pi) = \{x \in V : e(x, \pi) \text{ is minimum}\}$. The *median function* is the location function Med : $V^k \to 2^V \backslash \{\emptyset\}$ defined by $\text{Med}(\pi) = \{x \in V : s(x, \pi) \text{ is minimum}\}$.

A single-valued L is *onto* if, for any vertex v of G, there exists a profile π such that $L(\pi) = v$. A location function L is *unanimous* if, for each constant profile $(u, u, \ldots, u)$ on v consisting only of occurrences of the vertex u, we have $L((u, u, \ldots, u)) = \{u\}$.

The interpretation of a profile $(x_1, x_2, \ldots, x_k)$ is that x_j represents the most preferred location for voter j. Assuming that voter j wants to have the decision rule or location function lead to a choice of x_j or at least to include x_j in the set of chosen alternatives, how can a decision rule or location function prevent j from misrepresenting his or her true preference in order to gain an advantage? This is the intuitive notion of strategy-proofness and the following is an attempt to make this precise for location functions. Let $L : V^k \to 2^V \backslash \{\emptyset\}$ be a location function of order k on G. Then L is *strategy-proof* of the type SPi if, for $i \in \{1, 2, 3, 4\}$, L satisfies the following:

SP1: For every profile $\pi = (x_1, \ldots, x_k) \in V^k$, $j \in \{1, \ldots, k\}$ and $w \in V$,

$$d(x_j, L(\pi)) \leq d(x_j, L(\pi[x_j \to w])).$$

SP2: For every profile $\pi = (x_1, \ldots, x_k) \in V^k$ and $j \in \{1, \ldots, k\}$, if $x_j \notin L(\pi)$, then there does not exist a $w \in V$ such that $x_j \in L(\pi[x_j \to w])$.

SP3: For every profile $\pi = (x_1, \ldots, x_k) \in V^k$, if $x_j \in L(\pi)$ with $|L(\pi)| > 1$, then there does not exist a $w \in V$ such that $\{x_j\} = L(\pi[x_j \to w])$.

SP4: For every profile $\pi = (x_1, \ldots, x_k) \in V^k$ and $j \in \{1, \ldots, k\}$, if $x_j \notin L(\pi)$, then there does not exist a $w \in V, w \neq x_j$, such that $\{x_j\} = L(\pi[x_j \to w])$.

Clearly SP1 implies SP2 implies SP4.

Examples.

1. SP2 does not imply SP1: This example draws on ideas found in [11]. Let G be the path on three vertices denoted in order a_1, a_2, a_3, and let $L(\pi) = a_j$ where $a_j \in \{\pi\}$ appears most frequently in π and j is the smallest index among such vertices. Now let $\pi = (x_1, x_2, x_3) = (a_1, a_2, a_3)$. Then $L(\pi) = a_1$ and $d(x_3, L(\pi)) = 2$. But $d(x_3, L(\pi[x_3 \to a_2]) = d(a_3, L(a_1, a_2, a_2)) = d(a_3, a_2) = 1$.
2. SP4 does not imply SP2: We will show in Proposition 3 that Cen is such an example on the path on four vertices.

If L is single-valued then SP3 does not apply and SP2 and SP4 are equivalent. Also, when L is single-valued, SP1 corresponds to the definition found in [12]: voter j will never be able to improve (from her/his point-of-view) the result of applying the location function by reporting anything other than their peak choice x_j. SP2 implies that if voter j's top choice is not returned by L, then it cannot be made a part of the output set by j's reporting something else as top choice. SP3 requires that when j's top choice is returned by L along with others, this choice cannot be made into the unique element in the output set by reporting something else. Finally, SP4 says that when j's top choice is not returned by L, it cannot be the unique output returned by L if j reports a different choice.

The following result appears to be well-known [2, 12] but we include a proof for completeness since our context differs, as mentioned previously.

Lemma 1. *Let L be a single-valued location function of order k on G that satisfies SP1. Then L is onto if and only if L is unanimous.*

Proof. Clearly, a unanimous location function is onto.

Conversely assume that L is onto and let u be an arbitrary vertex of G. Because L is onto, there is a profile $\pi = (y_1, y_2, \ldots, y_k)$ with $L(\pi) = u$. Let $\rho = (x_1, x_2, \ldots, x_k)$ be the profile with $x_j = u$ for all j, and let $\pi_0 = \rho$. For $j = 1, 2, \ldots, k$, let $\pi_j = \pi_{j-1}[x_j \to y_j]$. Note that $\pi_k = \pi$. Since L satisfies SP1, we have

$$d(u, L(\pi_{j-1})) \leq d(u, L(\pi_j)),$$

for $j = 1, 2, \ldots, k$. Hence

$$d(u, L(\rho)) \leq d(u, L(\pi)) = 0,$$

and the proof is complete. □

3 Strategy-Proof Functions on Paths

We first consider the simplest situation: the graph is a path. This corresponds to the problem of locating a vertex along a single highway, or street, and is a fairly standard case to be considered [7, 8]. Let P be a path of length n. Without loss of generality we may assume that $V = \{0, 1, \ldots, n\}$ is the vertex set of P with the vertices on P numbered consecutively so that $P = 0 \to 1 \to \ldots \to n$. Note that $d(u, v) = |u - v|$ for $u, v \in V$.

We now consider single-valued location functions of order k on P.

Let G be the graph P^k, that is, the Cartesian product of k copies of P. Thus V^k is the vertex set of G, and two vertices $\pi = (x_1, \ldots, x_k)$ and $\rho = (y_1, \ldots, y_k)$ of G are adjacent if and only if there is exactly one i such that $|x_i - y_i| = 1$, and $x_j = y_j$ for all $j \neq i$. The distance function on G is given by

$$d(\pi, \rho) = \sum_{i=1}^{k} |x_i - y_i|$$

where $\pi = (x_1, \ldots, x_k)$ and $\rho = (y_1, \ldots, y_k)$ are vertices of G.

Clearly V is a linearly ordered set under $\leq$, the usual ordering on the natural numbers. This can be used to induce a partial ordering, which we also denote by $\leq$, on V^k as follows: for $\pi = (x_1, \ldots, x_k)$ and $\rho = (y_1, \ldots, y_k)$ in V^k define

$$\pi \leq \rho \text{ if and only if } x_i \leq y_i \text{ for all } 0 \leq i \leq k.$$

We denote the poset $(V^k, \leq)$ by $G_{\leq}$. Note that $\rho = (y_1, \ldots, y_k)$ *covers* $\pi = (x_1, \ldots, x_k)$ in $G_{\leq}$ if, for some i, we have $y_i - x_i = 1$ with $x_j = y_j$ for all $j \neq i$. Because we want to focus on the graph structure as well as the order, we use $G_{\leq}$ in the sequel.

A location function $L : V^k \to V$ is *isotone* on $G_{\leq}$ if, for any two vertices π and ρ of $G_{\leq}$, $\pi \leq \rho$ implies $L(\pi) \leq L(\rho)$.

Theorem 1. *Let L be a single-valued location function of order k on the path P of length n and let $G = P^k$. If L satisfies SP1, then L is isotone on $G_{\leq}$.*

Proof. First we prove that L is order preserving on each edge of the Hasse diagram of $G_{\leq}$. Let $\pi\rho$ be an edge in G with $\pi = (x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)$ and $\rho = (x_1, \ldots, x_{j-1}, x_j + 1, x_{j+1}, \ldots, x_k)$. Thus ρ covers π in $G_{\leq}$. We have to prove that $L(\pi) \leq L(\rho)$. For convenience we write $x = x_j$ and $x'_j = x_j + 1$.

Assume to the contrary that $L(\pi) > L(\rho)$. We consider three cases:

Case 1. $L(\pi) > L(\rho) \geq x + 1$.

Note that we can write $\rho = \pi[x_j \to x + 1]$. Since L satisfies SP1, this implies that

$$d(x_j, L(\pi)) \leq d(x_j, L(\pi[x_j \to x + 1]),$$

which can be written as

$$d(x, L(\pi)) \leq d(x, L(\rho)).$$

Due to the choice of V and the distance function d of P, this amounts to

$$L(\pi) - x \leq L(\rho) - x,$$

which is impossible.

Case 2. $x \geq L(\pi) > L(\rho)$.

Note that we can write $\pi = \rho[x'_j \to x]$. SP1 implies that

$$d(x'_j, L(\rho)) \leq d(x'_j, L(\rho[x'_j \to x]),$$

which can be written as

$$d(x+1, L(\rho)) \leq d(x+1, L(\pi)).$$

Due to the properties of the distance function d on P, this amounts to

$$x + 1 - L(\rho) \leq x + 1 - L(\pi),$$

which is impossible.

Case 3. $L(\pi) \geq x + 1 > x \geq L(\rho)$.

Note that we can write $\rho = \pi[x_j \to x + 1]$. Then SP1 implies that

$$d(x_j, L(\pi)) \leq d(x_j, L(\pi[x_j \to x + 1])),$$

which can be written as

$$d(x, L(\pi)) \leq d(x, L(\rho)).$$

Due to the properties of the distance function d on P this amounts to

$$L(\pi) - x \leq x - L(\rho).$$

Hence we have

$$L(\pi) + L(\rho) \leq 2x. \tag{1}$$

Now we write $\pi = \rho[x'_j \to x]$. Then SP1 gives that

$$d(x'_j, L(\rho)) \leq d(x'_j, L(\rho[x'_j \to x]),$$

which can be written as

$$d(x+1, L(\rho)) \leq d(x+1, L(\pi)).$$

This amounts to

$$x + 1 - L(\rho) \leq L(\pi) - (x + 1).$$

Hence we have

$$2(x + 1) \leq L(\pi) + L(\rho). \tag{2}$$

Clearly (1) and (2) yield a contradiction, which proves that L preserves order on the edges of $G_{\leq}$.

Now consider any two vertices π and ρ of $G_{\leq}$ with $\pi \leq \rho$. Since L is isotone on edges of $G_{\leq}$, it is isotone on all the edges in a shortest ordered path from π to ρ, which implies that $L(\pi) \leq L(\rho)$. □

The converse of Theorem 1 is not true, even if the isotone location function is onto.

Example. Define the *average function* A on P by $A(\pi) = \lfloor \frac{1}{k} \sum_{i=1}^{k} x_i \rfloor$, where $\pi = (x_1, \ldots, x_k)$. It is straightforward to check that the average function is an isotone, onto location function on $G_{\leq}$, but that it does not satisfy SP1. For a specific example, consider $\pi = (x_1, \ldots, x_k) = (0, 1, \ldots, 1, 1)$ and $\pi[x_k \to 2]$. Then $A(\pi) = (k-1)/k$ and $A(\pi[x_k \to 2]) = 1$ so $d(x_k, A(\pi)) > d(x_k, A(\pi[x_k \to 2]))$.

Theorem 2. *Let L be an onto single-valued location function on the path P of length n that satisfies SP1. Then*

$$\min_{x_j \in \pi}(x_j) \leq L(\pi) \leq \max_{x_j \in \pi}(x_j),$$

for any profile π on P.

Proof. Set $\alpha = \min_{x_j \in \pi}(x_j)$ and $\beta = \max_{x_j \in \pi}(x_j)$. By Lemma 1, we have $L((\alpha, \alpha, \ldots, \alpha)) = \alpha$ and $L((\beta, \beta, \ldots, \beta)) = \beta$. Then in $G_{\leq}$ there is an ordered path from $(\alpha, \alpha, \ldots, \alpha)$ to $(\beta, \beta, \ldots, \beta)$ passing through π. Since L satisfies SP1, the assertion now follows from Theorem 1. □

4 Strategy-Proofness of the Center Function

In this section we investigate how Cen behaves on paths, complete graphs, cycles, and graphs with diameter greater than 2. Let P_n denote the path $a_1 a_2 \cdots a_n$ with n vertices, and let K_n denote the complete graph on n vertices. Recall that the *diameter* of a graph G is the maximum $d(x, y)$ for $x, y \in V$. Since Cen is unanimous, trivially Cen satisfies SP1, SP2, SP3, and SP4 on $P_1 = K_1$.

Proposition 1. *Let $G = K_n$ and $k > 1$. Then* Cen *satisfies SP1, SP2, SP3, SP4 on G.*

Proof. If $\pi = (x_1, \ldots, x_k)$ is a profile with $|\pi| = 1$, we are done since Cen is unanimous. So assume $|\pi| > 1$. Then $\text{Cen}(\pi) = V$ and $\text{Cen}(\pi[x_j \to w]) = V$ or $\text{Cen}(\pi[x_j \to w]) = \{w\}$. SP1 holds since $d(x_j, V) = 0$, and therefore SP2 and SP4 hold. SP3 holds because if $|\text{Cen}(\pi[x_j \to w])| = 1$, then $\text{Cen}(\pi[x_j \to w) \neq \{x_j\}$. □

Proposition 2. *Let graph G have diameter at least 3 and $k > 1$. Then* Cen *violates conditions SP1 and SP2.*

Proof. Let $au_1u_2\cdots u_p$ be a shortest path of length at least 3 from a to u_p, so $p \geq 3$. Let $\pi = (x_1, \ldots, x_k) = (a, a, \ldots, a, u_2)$. Then $\text{Cen}(\pi) = \{v \in V : av \in E, u_2v \in E\}$. Now $\text{Cen}((\pi[x_k \to u_3]) = \text{Cen}((a, a, \ldots, u_3))$ contains u_1 and u_2. In particular, $x_k = u_2 \in \text{Cen}((\pi[x_k \to u_3])$ while $x_k \notin \text{Cen}(\pi)$. Thus, SP2 fails and therefore so does SP1. □

4.1 Paths

We now consider the center function on the path P_n of n vertices, which we will denote in order on the path as $a_1a_2\cdots a_n$. We may consider $n > 2$ since $n = 2$ gives us a complete graph and so here SP1 through SP4 hold by Proposition 1.

Proposition 3. *Suppose* Cen *is defined on* P_n *for* $n > 2$, *and let* $k > 1$. *Then*

1. Cen *satisfies SP1 if and only if* $n = 3$.
2. Cen *satisfies SP2 if and only if* $n = 3$.
3. Cen *fails SP3 for all* $n > 2$.
4. Cen *satisfies SP4 if and only if* $n \in \{3, 4\}$.

Proof. We first observe that SP3 fails for $n > 2$. If $\pi = (x_1, \ldots, x_k) = (a_1, a_1, \ldots, a_1, a_2)$, then $\text{Cen}(\pi) = \{a_1, a_2\}$. However, $\text{Cen}(\pi[x_k \to a_3]) = \text{Cen}((a_1, a_1, \ldots, a_1, a_3)) = \{a_2\}$, which contradicts condition SP3.

We next consider SP1, SP2, and SP4 for the case $n = 3$. It suffices to show that SP1 holds, for then SP2 and SP4 follow. Suppose that $d(x_j, \text{Cen}(\pi)) > d(x_j, \text{Cen}(\pi[x_j \to w])$. Because $n = 3$, $d(x_j, \text{Cen}(\pi))$ is equal to 1 or 2. If it is 2, then without loss of generality $x_j = a_1$ and $\text{Cen}(\pi) = \{a_3\}$, so $\{\pi\} = \{a_3\}$ and since Cen is unanimous SP1 cannot fail for this π. If $d(x_j, \text{Cen}(\pi)) = 1$, then $x_j \in \text{Cen}(\pi[x_j \to w])$. We may assume that $|\pi| > 1$, so without loss of generality, $\{\pi\} = \{a_1, a_2\}, \{a_1, a_3\}$, or $\{a_1, a_2, a_3\}$. Since $x_j \notin \text{Cen}(\pi)$, in the first case $x_j = a_3$, and in the second and third cases $x_j = a_1$ or a_3, without loss of generality the former. The first case is impossible since x_j must be in $\{\pi\}$. In the second and third cases, since $x_j = a_1$ is in $\text{Cen}(\pi[x_j \to w])$, we cannot have a_3 in $\{\pi[x_j \to w]\}$, which contradicts $\{\pi\} = \{a_1, a_3\}$ or $\{\pi\} = \{a_1, a_2, a_3\}$. We conclude that SP1 holds.

Suppose $n \geq 4$. By Proposition 2, SP1 and SP2 fail. Next consider $n \geq 5$ and let $\pi = (x_1, \ldots, x_k) = (a_1, a_1, \ldots, a_1, a_3)$. Then $\text{Cen}(\pi) = \{a_2\}$. However, $\text{Cen}(\pi[x_k \to a_5]) = \text{Cen}((a_1, a_1, \ldots, a_1, a_5)) = \{a_3\}$, so SP4 fails.

It is left to prove that SP4 holds for $n = 4$. Suppose that $\text{Cen}(\pi[x_j \to w]) = \{x_j\}$. Since $w \in \{\pi[x_j \to w]\}$ and $w \neq x_j$, we have $|\pi[x_j \to w]| > 1$. Since $\text{Cen}(\pi[x_j \to w])$ has only one element, this eliminates as $\{\pi[x_j \to w]\}$ all subsets of $\{a_1, a_2, a_3, a_4\}$ except for the four cases: $\{a_1, a_3\}, \{a_2, a_4\}, \{a_1, a_2, a_3\}$, $\{a_2, a_3, a_4\}$. By symmetry, we need only consider the first and the third. In both of these cases, $\text{Cen}(\pi[x_j \to w])$ is $\{a_2\}$, which means that $a_2 = x_j$ is also

in $\{\pi\}$. Thus, since $\{\pi[x_j \to w]\}$ is either $\{a_1, a_3\}, \{a_1, a_2, a_3\}$, $\{\pi\}$ is one of $\{a_1, a_2\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}$. In each case $x_j = a_2 \in \text{Cen}(\pi)$, which implies that SP4 holds. □

4.2 Cycles

We now consider Cen on the cycle C_n of n vertices, which we will denote in order on the cycle as $a_1, a_2, \ldots, a_n$. We may consider $n > 3$ since $n = 3$ gives K_3 and then, for $k > 1$, SP1 through SP4 hold by Proposition 1.

Proposition 4. *For a cycle C_n with $n > 3$:*

1. Cen *satisfies SP1 iff* $n = 4, k = 2; n = 4, k = 3;$ *or* $n = 5, k = 2$.
2. Cen *satisfies SP2 iff* $n = 4, k = 2; n = 4, k = 3;$ *or* $n = 5, k = 2$.
3. Cen *satisfies SP3 iff* $n = 4, k = 2$.
4. Cen *satisfies SP4 iff* $n = 4, k \geq 2; n = 5, k \geq 2; n = 6, k \geq 2; n = 7, k = 2;$ $n = 8, k = 2$.

Proof. Note that if $n \geq 6$, then C_n has diameter at least 3, so by Proposition 2, SP1 and SP2 fail. Now let $n = 4$ or 5. Suppose that $k \geq n$. Let $\pi = (a_1, a_1, a_2, a_3, \ldots, a_{n-1})$. Note that since $n - 1 \geq 3, a_1 \notin \text{Cen}(\pi)$. However, $a_1 \in \text{Cen}(\pi[x_1 \to a_n]) = V(G)$, and thus condition SP2, and hence also SP1, fails. For SP1 and SP2, this leaves the cases $n = 4, k = 2; n = 4, k = 3; n = 5, k = 2;$ $n = 5, k = 3; n = 5, k = 4$, which we consider next.

If $n = 4$ and $k \leq 3$, then up to symmetry, the only possibilities for $\{\pi\}$ that we need to consider are $\{a_1\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_1, a_2, a_3\}$. In the first case, since Cen is unanimous, SP1 is satisfied and thus so is SP2. In the second case, $\text{Cen}(\pi) = \{\pi\}$ so $d(x_j, \text{Cen}(\pi)) = 0$ so SP1 and therefore SP2 holds. In the third case, suppose without loss of generality that $j = 1$ and that $x_1 = a_1$. Then $d(x_j, \text{Cen}(\pi)) = d(x_j, \{a_2, a_4\}) = 1$. Since $k \leq 3$, the only possibility for $\{\pi\}$ is $\{a_1, a_3\}$. It follows that for $w \neq a_1, \{\pi[x_1 \to w]\}$ is either $\{a_3, w\}$ or $\{a_1, a_3, w\}$, and in each case a_1 is not in $\text{Cen}(\pi[x_1 \to w])$. Hence, SP1 holds and thus so does SP2. In the fourth case, up to interchange of order, $\pi = (a_1, a_2, a_3)$ since $k \leq 3$. Without loss of generality, $j = 1$ or $j = 2$. Suppose first that $j = 1$ and, without loss of generality, $x_1 = a_1$. Then $d(x_1, \text{Cen}(\pi)) = d(a_1, a_2) = 1$. Then $\{\pi[x_1 \to w]\} = \{a_2, a_3\}$ or $\{a_2, a_3, a_4\}$ and $a_1 \notin \text{Cen}(\pi[x_1 \to w])$, so $d(x_1, \text{Cen}(\pi[x_1 \to w])) \geq 1$. If $j = 2$, then $\{\pi[x_2 \to w]\} = \{a_1, a_3\}$ or $\{a_1, a_3, a_4\}$ and again a_1 is not in $\text{Cen}(\pi[x_2 \to w])$ and $d(x_2, \text{Cen}(\pi[x_2 \to w])) \geq 1$. This proves SP1 and thus SP2.

Next, let $n = 5, k = 2$. Then up to symmetry, $\pi = (a_1, a_1), (a_1, a_2)$, or (a_1, a_3) and we may take $x_1 = a_1$. In the first two cases, $d(x_1, \text{Cen}(\pi)) = 0$ and so SP1 and therefore SP2 holds. In the third case, $d(x_1, \text{Cen}(\pi)) = 1$ and $\text{Cen}(\pi[x_1 \to w]) = \{a_3\}, \{a_2, a_3\}$ or $\{a_3, a_4\}$. In every case, $a_1 \notin \text{Cen}(\pi[x_1 \to w])$ and so $d(x_1, \text{Cen}(\pi[x_1 \to w]) \geq 1$, which gives SP1 and thus SP2.

To complete the proof for SP1 and SP2, there are two more cases. First, let $n = 5, k = 3$. Take $\pi = (a_1, a_1, a_3)$. Then $x_1 = a_1 \notin \text{Cen}(\pi)$ but $x_1 = a_1 \in \text{Cen}(\pi[x_1 \to a_5]) = \text{Cen}(a_5, a_1, a_3) = V(G)$. Thus, SP2 fails and, therefore, SP1 fails. Next, let $n = 5, k = 4$. Take $\pi = (a_1, a_1, a_1, a_3)$. Then $x_1 = a_1 \notin \text{Cen}(\pi)$ but $x_1 \in \text{Cen}(\pi[x_1 \to a_5]) = V(G)$, so SP2 fails and therefore so does SP1.

Now consider SP3. Let $n \geq 4, k \geq 3$. Take $\pi = (a_1, a_1, \ldots, a_1, a_2)$. Then $\text{Cen}(\pi) = \{a_1, a_2\}$ but $\text{Cen}(\pi[x_1 \to a_n]) = \{a_1\}$, which shows that SP3 fails. Now let $n \geq 5, k = 2$. Let $\pi = (a_1, a_n)$. Then $\text{Cen}(\pi) = \{a_1, a_n\}$ but $\text{Cen}(\pi[x_1 \to a_2]) = \{a_1\}$, so SP3 fails. Finally, if $n = 4, k = 2$, suppose $\text{Cen}(\pi[x_1 \to w])$ has only one element, a_1. Since $n = 4, k = 2$, we must have $\pi = (a_1, a_1)$, which is impossible since $w \neq x_1$. Thus, SP3 holds.

Finally, consider SP4. First, suppose $n = 4$. If $\text{Cen}(\pi[x_j \to w])$ has one element x_j, then without loss of generality $\{\pi[x_j \to w]\} = \{a_1\}$ or $\{a_1, a_2, a_3\}$. The former case is impossible since $x_j = a_1$ and $w \neq x_j$ must both be in $\{\pi[x_j \to w]\}$. In the latter case, $\text{Cen}(\pi[x_j \to w]) = \{a_2\}$ and $\{\pi\} = \{a_1, a_2\}, \{a_2, a_3\}, \{a_1, a_3\}$, or $\{a_1, a_2, a_3\}$. In each case, $a_2 \in \text{Cen}(\pi)$, which implies that SP4 holds.

Next, let $n = 5$. If $\text{Cen}(\pi[x_j \to w])$ has one element x_j, then without loss of generality $\{\pi[x_j \to w]\} = \{a_1\}, \{a_1, a_3\}$ or $\{a_1, a_2, a_3\}$. The former case is impossible as with $n = 4$. In the other two cases, $\{a_2\} = \text{Cen}(\pi[x_j \to w])$ and $x_j = a_2$. If $\{\pi[x_j \to w]\} = \{a_1, a_3\}$, then $\{\pi\} = \{a_1, a_2\}, \{a_2, a_3\}$, or $\{a_1, a_2, a_3\}$. In each case, $a_2 \in \text{Cen}(\pi)$, which implies that SP4 holds. If $\{\pi[x_j \to w]\} = \{a_1, a_2, a_3\}$, then we have the same possible sets $\{\pi\}$ and again we get SP4.

Suppose $n = 6$. If $\text{Cen}(\pi[x_j \to w])$ has one element x_j, then without loss of generality $\{\pi[x_j \to w]\} = \{a_1\}, \{a_1, a_3\}, \{a_1, a_2, a_3\}$, or $\{a_1, a_2, a_3, a_4, a_5\}$. The first three cases are handled as for $n = 5$. In the fourth case, $\text{Cen}(\pi[x_j \to w]) = \{a_3\}$. Now $\{\pi\}$ has to be one of the sets $\{a_1, a_2, a_3, a_4, a_5\}, \{a_1, a_2, a_3, a_4\}, \{a_1, a_2, a_3, a_5\}, \{a_1, a_3, a_4, a_5\}, \{a_2, a_3, a_4, a_5\}$. Since $a_3 \in \text{Cen}(\pi)$ in all cases, SP4 holds.

Next, take $n = 7$. When $k \geq 4$, consider $\pi = (a_1, a_1, \ldots, a_1, a_2, a_3)$. Then $\text{Cen}(\pi) = \{a_2\}$. Now $\text{Cen}(\pi[x_{k-2} \to a_6]) = \text{Cen}((a_1, a_1, \ldots, a_1, a_6, a_2, a_3)) = \{a_1\}$, so SP4 fails. (Note that $k \geq 4$ is used since it implies that $k - 2 \geq 2$ and thus $\{\pi\}$ has a_1 in it.) When $k = 3$, consider $\pi = (a_1, a_2, a_3)$, with $\text{Cen}(\pi) = \{a_2\}$. Then $\text{Cen}(\pi[x_3 \to a_5]) = \text{Cen}((a_1, a_2, a_5)) = \{a_3\}$, so SP4 fails. Suppose next that $k = 2$ and that $\{\pi[x_j \to w]\} = \{x_j\}$. Since $k = 2$, without loss of generality $\pi[x_j \to w] = (a_1, a_1)$ or (a_1, a_3). In the former case, $x_j = a_1$ is in $\text{Cen}(\pi)$. In the latter case, $x_j = a_2$ and $\{\pi\} = \{a_1, a_2\}$ or $\{a_2, a_3\}$, so $x_j \in \text{Cen}(\pi)$ and SP4 holds.

To handle the case $n = 8$, suppose first that $k \geq 4$, and consider $\pi = (a_1, a_1, \ldots, a_1, a_2, a_3)$. (As in the case $n = 7$, the assumption $k \geq 4$ is used.) Then $\text{Cen}(\pi) = \{a_2\}$. Now $\text{Cen}(\pi[x_k \to a_5]) = \text{Cen}((a_1, a_1, \ldots, a_1, a_2, a_5)) = \{a_3\}$, so SP4 fails. When $k = 3$, the same example as with $n = 7$ shows that SP4 fails. Finally, take $k = 2$. That SP4 holds follows in the same way as with $n = 7$.

To conclude the proof, consider $n \geq 9$. Take $\pi = (a_1, a_1, \ldots, a_1, a_3)$. Note that $\text{Cen}(\pi) = \{a_2\}$, but $\text{Cen}(\pi[x_k \to a_5]) = \text{Cen}((a_1, a_1, \ldots, a_1, a_5\}) = \{a_3\}$, so SP4 fails. □

5 The Median Function on Median Graphs

We now study how the median function behaves on median graphs with respect to strategy-proofness. Median graphs form a class of bipartite graphs that include trees and n-cubes. Specifically, a *median graph* is a connected graph $G = (V, E)$ such that for every three vertices $x, y, z \in V$, there is a unique vertex w on a shortest-length path between each pair of x, y, z. Let $I(x, y) = \{w \in V : d(x, w) + d(w, y) = d(x, y)\}$. Then it is easy to see that G is a median graph if and only if $|I(x, y) \cap I(x, z) \cap I(y, z)| = 1$ for all $x, y, z \in V$.

First we present some necessary concepts and results for arbitrary graphs. Then we concentrate on median graphs and recapitulate some necessary notation and results from [3,4,9,10].

Let $G = (V, E)$ be a connected graph. A subgraph H of G is *convex* if, for any two vertices x and y of H, all shortest x, y-paths lie completely in H. Note that convex subgraphs are induced. A subset W of V is *convex* if it induces a convex subgraph. A subgraph H is *gated* if, for any vertex w there exists a unique vertex x in H such that for each vertex y of H there exists a shortest w, y-path through x. This vertex x is the *gate* for w in H. Clearly, if H is gated, then the gate for w in H is the vertex of H closest to w. It is also the unique vertex z in H such that any shortest w, z-path intersects H only in w. A gated subset of vertices is a subset that induces a gated subgraph. Note that gated subgraphs are convex, but the converse need not be the case. A simple consequence of the theory on median graphs is that convex sets in a median graph are always gated. Let π be a profile on the median graph G and $uv \in E$. By W_{uv} we denote the subset of V of all vertices closer to u than to v, by G_{uv} the subgraph induced by W_{uv}. The subgraphs G_{uv}, G_{vu} form a so-called *split*: the sets W_{uv}, W_{vu} are disjoint with V as their union. We call G_{uv} and G_{vu} *split-sides*. Split-sides are convex subgraphs, and hence gated.

Let π be a profile, π_{uv} be the subprofile of π consisting of the vertices in π closer to u than v, and let $l(\pi_{uv})$ denote the number of terms in the sequence π_{uv}. Theorem 3 of [4] tells us that, for any profile π and any edge uv with $l(\pi_{uv}) > l(\pi_{vu})$ we have $\text{Med}(\pi) \subseteq G_{uv}$. An important consequence of this theorem is that

$$\text{Med}(\pi) = \bigcap_{l(\pi_{uv}) > l(\pi_{vu})} G_{uv}.$$

Since the intersection of convex subgraphs is again convex, median sets of profiles are thus convex, and hence also gated.

For any two vertices u, v in G the set of neighbors of u in $I(u, v)$ is denoted by $N_1(u, v)$. Loosely speaking these are precisely the vertices that are one step closer to v from u. Let $G_{x/v} = \bigcap_{u \in N_1(v,x)} G_{vu}$, which signifies all vertices that are "behind" v seen from x, that is, all vertices that can be reached from x by a shortest path passing through v.

Lemma 2. *Let x and v be vertices in a median graph G. Then v is the gate for x in* $\bigcap_{u \in N_1(v,x)} G_{vu}$.

Proof. Since split-sides are convex, the subgraph $G_{x/v} = \bigcap_{u \in N_1(v,x)} G_{vu}$ is convex and hence gated. By definition, any shortest x, v-path intersects $G_{x/v}$ only in v. So indeed v is the gate for x in this subgraph. □

Corollary 1. *Let $\pi = (x_1, x_2, \ldots, x_k)$ be a profile on a median graph G. If x_j is not in* Med(π), *and m is the gate of x_j in* Med(π), *then* Med($\pi[x_j \rightarrow w]$) *is contained in $G_{x_j/m}$.*

Proof. First we show that Med(π) lies in $G_{x_j/m}$. Let u be any neighbor of m in $I(x_j, m)$. Then u is not in Med(π), so a majority of π lies in G_{mu}, whence Med(π) lies in G_{mu}, and we are done.

Now we replace x_j by w, thus obtaining the profile $\rho = \pi[x_j \rightarrow w]$. Take a neighbor u of m in $I(x, m)$. Note that a majority of π lies in G_{mu} and a minority lies in G_{um}, and x_j belongs to this minority. So, no matter where w is located, a majority of ρ still lies in G_{mu}. Hence Med(ρ) is contained in G_{mu}. This settles the proof. □

Theorem 3. *Let G be a median graph. Then* Med : $V^k \rightarrow 2^V \setminus \{\emptyset\}$ *satisfies SP1 (and therefore SP2 and SP4) for any k.*

Proof. Let $\pi = (x_1, x_2, \ldots, x_k)$ be a profile on G such that x_j is not in Med(π), and let w be any vertex of G. Let m be the gate of x_j in Med(π). Note that in G, $d(x_j, \text{Med}(\pi)) = d(x_j, m)$. By Corollary 1, Med($\pi[x_j \rightarrow w]$) lies in $G_{x_j/m}$. So each vertex y in Med($\pi[x_j \rightarrow w]$) can be reached from x_j via a shortest path passing through m. Hence $d(x_j, m) \leq d(x_j, y)$ for all $y \in \{\pi[x_j \rightarrow w]\}$, and we are done. □

6 Conclusions and Future Work

This note has introduced four notions of strategy-proofness and illustrated them for several location functions and for several types of graphs. We have only begun to investigate this subject and, even for this relatively small beginning, have left open questions to be addressed.

For instance, we have given an example of a function (the average function) that is an isotone, onto location function but does not satisfy SP1. We believe that under certain conditions, the converse holds, but leave the investigation of such conditions to future work.

Proposition 2 shows that for every graph of diameter at least 3, when $k > 1$, Cen violates SP1 and SP2. We have left open the question of whether this is also true of SP3 and SP4.

Section 5 shows that SP1, and therefore SP2 and SP4, hold for median graphs. It leaves open this question for SP3.

Section 4 determines the cases where SP1 through SP4 hold for the center function on paths and cycles. For the median function, since a path is a median graph, Sect. 5 handles SP1, SP2, and SP4. SP3 remains open. We have not attempted to categorize when these conditions of strategy-proofness hold for cycles. For trees, the fact that they are median graphs shows that SP1, SP2, and SP4 hold for the median function. SP3 remains open. For the center function, the case of trees other than paths remains an area for future research.

Acknowledgements Fred Roberts thanks the National Science Foundation for support under grant SES-1024722 to Rutgers University and the Department of Homeland Security for support under award 2009-ST-061-CCI002-04 to Rutgers University.

References

1. Alon, N., Feldman, M., Procaccia A.D., Tennenholtz, M.: Strategyproof approximation of the minimax on networks. Math. Oper. Res. **35**, 513–526 (2010)
2. Danilov, V.I.: The structure of non-manipulable social choice rules on a tree. Math. Soc. Sci. **27**, 123–131 (1994)
3. Klavžar, S., Mulder, H.M.: Median graphs: characterizations, location theory, and related structures. J. Combin. Math. Combin. Comput. **30**, 103–127 (1999)
4. McMorris, F.R., Mulder, H.M., Roberts, F.S.: The median procedure on median graphs. Discrete Appl. Math. **84**, 165–181 (1998)
5. McMorris, F.R., Roberts, F.S., Wang, C.: The center function on trees. Networks **38**, 84–87 (2001)
6. McMorris, F.R., Mulder, H.M., Powers, R.C.: The median function on distributive semilattices. Discrete Appl. Math. **127**, 319–324 (2003)
7. Miyagawa, E.: Locating libraries on a street. Soc. Choice Welf. **18**, 527–541 (2001)
8. Moulin, H.: On strategy proofness and single peakedness. Public Choice **35**, 437–455 (1980)
9. Mulder, H.M.: The structure of median graphs. Discrete Math. **24**, 197–204 (1978)
10. Mulder, H.M.: The Interval Function of a Graph, Mathematical Centre Tracts 132. Mathematisch Centrum, Amsterdam (1980)
11. Sanver, M.R.: Strategy-proofness of the plurality rule over restricted domains. Econ. Theory **39**, 461–471 (2009)
12. Schummer, J., Vohra, R.V.: Strategy-proof location on a network. J. Econ. Theory **104**, 405–428 (2002)
13. Taylor, A.D.: Social Choice and the Mathematics of Manipulation. Cambridge University Press, Cambridge (2005)

A Pseudo-Boolean Approach to the Market Graph Analysis by Means of the p-Median Model

Boris Goldengorin, Anton Kocheturov, and Panos M. Pardalos

Abstract In the course of recent 10 years algorithms and technologies for network structure analysis have been applied to financial markets among other approaches. The first step of such an analysis is to describe the considered financial market via the correlation matrix of stocks prices over a certain period of time. The second step is to build a graph in which vertices represent stocks and edge weights represent correlation coefficients between the corresponding stocks. In this paper we suggest a new method of analyzing stock markets based on dividing a market into several substructures (called stars) in which all stocks are strongly correlated with a leading (central, median) stock. Our method is based on the p-median model a feasible solution to which is represented by a collection of stars. Our reduction of the adjusted p-Median Problem to the Mixed Boolean pseudo-Boolean Linear Programming Problem is able to find an exact optimal solution for markets with at most 1,000 stocks by means of general purpose solvers like CPLEX. We have designed and implemented a high-quality greedy-type heuristic for large-sized (many thousands of stocks) markets. We observed an important "median nesting"

B. Goldengorin (✉)
Operations Department, Faculty of Economics and Business, University of Groningen, Nettelbosje 2, 9747 AE, Groningen, The Netherlands

Center of Applied Optimization, University of Florida, 401 Weil Hall, P.O. Box 116595, Gainesville, FL 32611-6595, USA
e-mail: b.goldengorin@rug.nl; goldengorin@ufl.edu

A. Kocheturov
Laboratory of Algorithms and Technologies for Network Analysis, National Research University Higher School of Economics, 136 Rodionova, Nizhny Novgorod, Russian Federation
e-mail: antrubler@gmail.com

P.M. Pardalos
Center of Applied Optimization, University of Florida, 401 Weil Hall, P.O. Box 116595, Gainesville, FL 32611-6595, USA
e-mail: pardalos@ufl.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_5,

property of returned solutions: the p leading stocks, or medians, of the stars are repeated in the solution for $p + 1$ stars. Moreover, many leading stocks (medians), for example, in the USA stock market are the well-known market indices and funds such as the Dow Jones, S&P which form the largest stars (clusters).

Keywords Stock markets analysis • Russia • Sweden • USA • p-Median problem • Pseudo-Boolean approach • Cluster analysis by stars • Leading stocks

1 Introduction

The main goal of the paper is to introduce a new method to analyze financial markets based on the correlation matrix of stock prices. Mantegna [1] suggested constructing the Minimal Spanning Tree (MST) which connects all stocks in a certain portfolio. He used a simple nonlinear transformation of the correlation matrix where the correlation coefficient p_{ij} between prices of stocks i and j is substituted by the number $d_{ij} = \sqrt{2(1 - p_{ij})}$. This number can be used as a distance measure between stock prices (and between stocks in general). Now this method is a base for many other variations of this method which modify it in several directions [2–8]. These modifications suggest other network structures instead of the MST including different metrics (measures).

Boginski et al. [9, 10] suggested to find large highly correlated groups of stocks in a market graph. The market graph is constructed as follows: each stock is represented by a vertex and two vertices are connected by an edge if the correlation coefficient of the corresponding pair of stocks (calculated over a certain period of time) exceeds a pre-specified threshold $\theta \in [-1; 1]$. The authors search for cliques and independent sets in the graph and highlight a special economic meaning of the maximum clique (which is almost a binary equivalence relation) and the maximum independent set of this graph. In a clique the behavior of all its stocks is similar but any choice of a single stock acting as a typical representative for all stocks in a clique is questionable. The p-Median Problem (PMP)-based approach returns stars which are natural relaxations of cliques and a typical (leading) stock is defined by its highest degree connections with other stocks within each star.

A number of recent publications are related to analyzing a financial market of a certain country [11–14]. The main goal of these papers is to find network structures which can describe the structure of a state-related market.

Our method consists in dividing all stocks presented on a market into several groups such that the stock prices are strongly correlated. For this purpose we calculate a correlation matrix $\mathrm{P} = \left[\rho_{ij}\right]_{n \times n}$ of prices for all n stocks on the given stock market and then use these obtained correlations as input data for further clustering (see the next section). The main idea of the clustering is to find a set S of stocks (hereinafter we call this set medians or leading stocks) with the predefined number p of leading stocks maximizing the total "similarity" over all stocks clustered by means of p leading stocks. By a "similarity" $\rho(i, S)$ between

the i-th stock and the set S we mean a maximum correlation between a price of this stock and prices of all stocks in the set $|S| = p$: $\rho(i, S) = \max\limits_{j \in S}(\rho_{ij})$. We do not use any threshold and thus do not lose any distinctions between the tightness of stocks within a cluster (star) to the leading stock. For example, if the number of stocks included in the returned cluster i (star) is equal to k_i for all $i \in S$ (in this notation every cluster is identified by the median), then the *average tightness* T_i of all stocks within a cluster to the leading stock i we define as follows: $T_i = (\sum_{j \in K_i} \rho_{ij})/k_i$.

After introducing the "similarity" $\rho(i, S)$ we move to solving the following problem:

$$\max_{S \subset X, |S|=p} \sum_{i=1}^{n} \rho(i, S) = \max_{S \subset X, |S|=p} \left(\sum_{i=1}^{n} \max_{j \in S}(\rho_{ij}) \right), \tag{1}$$

where X is a set of all stocks on the market, n is the number of stocks, and p is the number of clusters. Applying the following transformation of the entries $\rho_{ij} \geq 0$ in the correlation matrix P to a complementary matrix $C = \left[c_{ij}\right]_{n \times n} = \left[1 - \rho_{ij}\right]_{n \times n}$ we get an equivalent objective for (1) as follows:

$$\min_{S \subset X, |S|=p} \left(\sum_{i=1}^{n} \min_{j \in S}(c_{ij}) \right). \tag{2}$$

We use the capital C to mark this complementary matrix because it can be considered as a distance matrix between all stocks. Notice that there are several ways how to obtain such a distance matrix.

Formula (2) corresponds to the combinatorial optimization formulation of the PMP where the set of potential locations and the set of customers (clients) are the same. In this paper both of these sets, namely locations and customers, are stocks. A detailed overview of the PMP can be found in Reese [15] and Mladenovic et al. [16]. For solving this problem we apply the Pseudo-Boolean approach originally introduced by Hammer [17] and Beresnev [18] which further developed and applied in AlBdaiwi et al. [19, 20], Goldengorin and Krushinsky [21], Goldengorin et al. [22, 23]. This approach gives us the most compact formulation of the Mixed Integer Linear Programming (MILP) model. The objective function of MILP formulation is a linearized pseudo-Boolean polynomial (pBp). The MILP formulation for small financial markets such as Russian and Sweden can be easily solved by means of a general purpose MILP solver like CPLEX or LPSolve. Large financial markets (e.g., the USA market with 3,378 stocks) cannot be clustered within half an hour CPU time by means of these solvers. So we apply an efficient greedy heuristic which returns high quality clusters.

2 Construction of the Correlation Matrix

In this paper we analyze the financial market of the USA, the biggest in the world, Russian market as a representative of the developing markets, and Swedish market due to the fact that Sweden is one of the developed countries with a very high income per person. We take Russian market's data from an open source, a website of the "Finam" investment company www.finam.ru. There are about 700 Russian issuers traded on the Moscow Interbank Currency Exchange (MICEX) but we take into consideration only those stocks which had been traded at least 80 % of all trading days from September 3, 2007 till September 16, 2011. This period of time includes 1,000 trading days and 151 companies. In Swedish stock market we take 266 stocks for the same period of time and with the same requirements. In American stock market we take 3,378 financial instruments traded for the same period of time and satisfying the same requirements from about 7,000 companies shares and stock market indices. We obtain data for both these markets with Application Programming Interface (sometimes cited as API) at the website www.finance.yahoo.com.

In order to calculate the correlation matrices for the markets we use the following formula [10, 14]:

$$\rho_{ij} = \frac{E\left\{\left(R_i - E\left\{R_i\right\}\right)\left(R_j - E\left\{R_j\right\}\right)\right\}}{\sqrt{\operatorname{var}\left(R_i\right)\operatorname{var}\left(R_j\right)}}, \tag{3}$$

which gives the correlation coefficient between prices of two stocks i and j. R_i is a new time series obtained from the original one according to the following rule: $R_i\ (t) = \ln \frac{P_i(t)}{P_i(t-1)}$, $P_i\ (t)$ is a closure price of the financial instrument i at the day t.

The final step consists in obtaining a complementary matrix from the correlation matrix in order to cluster the market by means of the PMP.

3 The PMP and the Pseudo-Boolean Approach

In this section we show how to obtain a MILP formulation of the model (2).

Let us consider a simple market with four stocks and positive correlation coefficients computed according to the formula (3). The complementary of the correlation matrix is listed below:

$$C = \begin{bmatrix} 0 & 0.2 & 0.6 & 0.7 \\ 0.2 & 0 & 0.8 & 0.5 \\ 0.6 & 0.8 & 0 & 0.1 \\ 0.7 & 0.5 & 0.1 & 0 \end{bmatrix} \tag{4}$$

The idea of clustering is to choose p medians which satisfy the model (2). In this case the found medians will be the centers of clusters (leading stocks) and inside each cluster all other stocks will strongly correlate with the corresponding median (leading) stock. Applying the procedure introduced originally by AlBdaiwi et al. [19] and developed by Goldengorin and Krushinsky [21], we get a pseudo-Boolean polynomial (pBp):

$$\begin{aligned} f(C) = {} & 0.2y_1 + 0.2y_2 + 0.1y_3 + 0.1y_4 + 0.7y_1y_2 + 0.9y_3y_4 \\ & + 0.1y_1y_2y_3 + 0.3y_1y_2y_4 + 0.2y_1y_3y_4 + 0.2y_2y_3y_4 \end{aligned} \tag{5}$$

where y_i is equal to 0 if the stock i is a median (leading) stock and 1, otherwise.

Now we can formulate the problem of clustering the market in terms of the MILP model.

The pBp can be truncated (see AlBdaiwi et al. [19]) due to the fact that all monomials (terms) in the pBp with degree (by the degree we mean a number of multiplied Boolean variables in the monomial) k bigger than $n - p$ can be removed from the pBp (see for more details in Goldengorin et al. [23]). Indeed, according to the constraint $\sum_{i=1}^{n} y_i = n - p$ we have only $n - p$ nonzero variables and every monomial with the degree $k > n - p$ includes at least $k - (n - p) > 0$ zero variables. Thus such monomials are equal to zero and we can remove them from the original pBp. For example, let $p = 2$. We can remove from our polynomial $f(C)$ in (5) all terms with the degree $k > 2$. AlBdaiwi et al. [19] have termed the obtained polynomial $f_t(C)$ as *a truncated polynomial*:

$$f_t(C) = 0.2y_1 + 0.2y_2 + 0.1y_3 + 0.1y_4 + 0.7y_1y_2 + 0.9y_3y_4 \tag{6}$$

In the next section we show how to incorporate the linearized truncated polynomial $f_t(C)$ into a MILP model.

4 Preprocessing and Exact Model

In order to create a MILP model of the original PMP we show how AlBdaiwi et al. [19] have linearized the truncated pBp by introducing new continuous nonzero decision variables which in fact will take only Boolean values $\{0, 1\}$. In the second row of Table 1 the number of entries in C is indicated while the third row of Table 1 shows the number of terms in pBp after elimination the equal entries and the fourth row of Table 1 shows the number of terms in pBp after aggregation the similar terms and finally the number of terms in a truncated pBp depending on the number p indicated in rows $p = 2, \ldots, 15$. There is an essential distinction between entries of correlation matrices for Russian and Sweden markets on one side and USA market on the other side. The American market has more statistically similar correlation dependencies between different stocks compared to the stocks

Table 1 Reduction of the pBp for the financial markets

Country	Russia	Sweden	USA
# of stocks	151	266	3379
# of entries in C	22801	70756	11417641
$\#T$	22768	70582	7943904
$\#T_r$	22292	69874	7739667
Reduction (%)	2.23235823	1.246537396	32.21308149
$p = 2$	22251	69802	7739533
$p = 3$	22147	69602	7739181
$p = 4$	22012	69357	7738620
$p = 5$	21865	69096	7737856
$p = 6$	21716	68831	7736944
$p = 7$	21568	68566	7735935
$p = 8$	21419	68301	7734867
$p = 9$	21268	68036	7733742
$p = 10$	21117	67772	7732512
$p = 11$	20966	67506	7731239
$p = 12$	20815	67241	7729880
$p = 13$	20664	66975	7728474
$p = 14$	20513	66709	7727102
$p = 15$	20362	66443	7725750

in Russian and Sweden markets. It means that the American stock market is more stable and influential compared to Russian and Sweden markets.

A term t with degree k can be substituted by a continuous variable z with two constraints: $z \geq 0$ and $z \geq \sum_{i=1}^{k} y_{h_i} - k + 1$ where $t = \prod_{i=1}^{k} y_{h_i}$. These constraints together with the objective to be minimized guarantee that the continuous variable $z = 1$ if and only if all Boolean variables $y_{h_i} = 1$. In all other cases, $z = 0$. For instance, two terms $t_1 = y_1 y_2$ and $t_2 = y_3 y_4$ of the polynomial (5) can be substituted by two continuous z_1 and z_2 and the linearized formulation of the MILP model with the objective function (6) is:

$$
\begin{aligned}
& p_t(C) = 0.2y_1 + 0.2y_2 + 0.1y_3 + 0.1y_4 + 0.7z_1 + 0.9z_2 \rightarrow \min, \\
& s.t.: \\
& z_1 \geq y_1 + y_2 - 1; \\
& z_1 \geq 0; \\
& z_2 \geq y_3 + y_4 - 1; \\
& z_2 \geq 0; \\
& y_1, y_2, y_3, y_4 \in \{0, 1\}
\end{aligned}
\tag{7}
$$

The MILP (7) now can be solved by means of any general purpose solver like CPLEX or LPSolve. We are interested only in y_i values which we take from the solution: if $y_i = 0$, the stock i is a median. The last step is to cluster stocks: for each stock we choose a median with the biggest correlation and move this stock to the cluster of this median.

5 Greedy Heuristics

For large financial markets such as the USA market the MILP model cannot be solved by means of CPLEX or LPSolve solvers within reasonable CPU times. In order to cluster such markets we introduce a new greedy-like heuristic which deals with the truncated pBp.

After the truncated pBp is obtained our goal is to set exactly p Boolean variables y_i to zero, so that the value of the pBp is minimized. When all the y_i are equal to 1 the pBp has its maximal value. When we set, for example, $y_1 = 0$, we remove all monomials which contain y_1 and the value of the pBp is reduced by the sum of the coefficients of these monomials. For example, for the pBp in formula (6) when we set y_1 or y_2 to zero we reduce the value of the pBp by $0.2 + 0.7 = 0.9$ and when we set y_3 or y_4 to zero we reduce the value of the pBp by $0.1 + 0.9 = 1.0$. In our heuristic we follow a greedy approach. First we set to zero the variable for which this reduction is maximal (the sum of the coefficients of the terms containing this variable is maximal). If there are several such variables, we choose the variable with the smallest index i. Then we remove the monomials containing this variable and again search for the variable with the maximum reduction of the pBp value. The greedy heuristic algorithm consists of the following steps:

Step 1. Calculate contributions of all y_i to the pBp: sum up coefficients of all monomials which include y_i, $\forall i = 1, \ldots, n$.

Step 2. Move to the set of medians S the y_i which gives the biggest contribution to the polynomial. If $|S| = p$, go to step 4. If there are several variables which have the biggest contribution, choose the first one.

Step 3. Remove from the pBp all monomials which include y_i with the biggest contribution. Go to step 1.

Step 4. Cluster the stocks. End.

We tested our greedy heuristics on benchmark instances of the Uncapacitated PMP taken from J.E. Beasley operational research library (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html). The results are presented in Table 2. Every problem has its own name, for instance, "pmed1." It is the first column of the table. It has also the number p of clusters (which is not shown in the table but can be found in the website) and the optimal objective function value (5) which is in the second column. The third column shows the heuristic objective function value. And the last column indicates a relative error of the heuristic solution calculated according to the formula $\text{Error} = 100\,\%(\text{Greedy}/\text{Exact} - 1)$.

Notice that despite the fact that we use the set $Y = \{y_1, y_2, \ldots, y_n\}$ of Boolean variables we have applied standard operations of addition and multiplication defined on the domain of the real numbers. It implies that the objective function values are real numbers. Due to that fact Hammer [17] termed such polynomials the *pseudo-Boolean polynomials*.

Also we compare our computational results for Russian market by means of our exact procedure and greedy heuristic (see Table 3).

Table 2 The greedy heuristic result on OR-Library instances

Name	Exact	Greedy	Error (%)	Name	Exact	Greedy	Error (%)
pmed1	5819	5819	0	pmed21	9138	9138	0
pmed2	4093	4118	0.6108	pmed22	8579	8605	0.3031
pmed3	4250	4272	0.5176	pmed23	4619	4629	0.2165
pmed4	3034	3046	0.3955	pmed24	2961	2982	0.7092
pmed5	1355	1378	1.6974	pmed25	1828	1866	2.0788
pmed6	7824	7943	1.521	pmed26	9917	9917	0
pmed7	5631	5646	0.2664	pmed27	8307	8364	0.6862
pmed8	4445	4462	0.3825	pmed28	4498	4518	0.4446
pmed9	2734	2771	1.3533	pmed29	3033	3070	1.2199
pmed10	1255	1274	1.5139	pmed30	1989	2020	1.5586
pmed11	7696	7721	0.3248	pmed31	10086	10086	0
pmed12	6634	6649	0.2261	pmed32	9297	9319	0.2366
pmed13	4374	4391	0.3887	pmed33	4700	4732	0.6809
pmed14	2968	2987	0.6402	pmed34	3013	3058	1.4935
pmed15	1729	1743	0.8097	pmed35	10400	10406	0.0577
pmed16	8162	8194	0.3921	pmed36	9934	9952	0.1812
pmed17	6999	6999	0	pmed37	5057	5075	0.3559
pmed18	4809	4851	0.8734	pmed38	11060	11141	0.7324
pmed19	2845	2889	1.5466	pmed39	9423	9423	0
pmed20	1789	1839	2.7949	pmed40	5128	5163	0.6825

Table 3 Comparison of the greedy and exact PMP solutions on Russian market for different number of clusters

p	Exact	Greedy	Error (%)	p	Exact	Greedy	Error (%)
1	99.8953702	99.8953702	0	16	79.15926	79.29346	0.169532
2	95.69383	95.69383	0	17	78.28544	78.42179	0.17417
3	93.615102	93.615102	0	18	77.41961	77.56736	0.190843
4	91.913129	91.913129	0	19	76.55851	76.715	0.204406
5	90.468442	90.468442	0	20	75.70408	75.86357	0.210676
6	89.101039	89.216384	0.129454	21	74.85172	75.01556	0.218886
7	87.939338	88.072194	0.151077	22	74.00371	74.18343	0.242853
8	86.823678	86.969944	0.168463	23	73.17158	73.35198	0.246544
9	85.757218	85.897973	0.164132	24	72.34013	72.52165	0.250926
10	84.72569	84.85525	0.152917	25	71.51073	71.69225	0.253836
11	83.71731	83.83751	0.143578	26	70.6814	70.86389	0.258187
12	82.78033	82.90053	0.145204	27	69.85814	70.03966	0.259841
13	81.84857	81.96362	0.140564	28	69.03577	69.21729	0.262936
14	80.93123	81.05646	0.154736	29	68.21529	68.39681	0.266099
15	80.03429	80.16783	0.166853	30	67.40807	67.57878	0.253249

The results obtained by the greedy heuristic are very close to the exact ones. Thus we can use our greedy heuristic as a powerful tool for large markets clustering. It allows us to divide the USA market into clusters in at most half an hour.

6 Results and Their Interpretation

Our computational results show that the network structure of financial markets has a property of "median nesting." This property means the following: if we find medians for p and $p + 1$ clusters, then almost all medians calculated for p clusters can be found among the medians calculated for $p + 1$ clusters. Moreover in case of financial markets all medians for p can be found among medians for $p + 10$. From the computational point of view this property gives us a new tool: first we can solve the problem instance by our either exact or greedy heuristic for a number of clusters big enough to provide an optimal (high quality) solution (for instance, $2p$) and then find p medians among already found ones.

We also found a "stable" number of clusters for Russian, Swedish, and American markets (see Table 4). If we cluster a market and there are no trivial clusters (clusters which include only one stock—the median) we say that this is a reasonable clustering. For all markets involved in our computational experiments there exist a number of clusters such that for all numbers less than this number we obtain a reasonable clustering and for all greater numbers the trivial clusters appear. So we believe that these "stable" numbers reflect the best clustering of the markets. The medians corresponding to the "stable" clustering for Swedish markets are presented in Table 5.

We also compared our results with the approach of Boginski et al. [9, 10] of finding the maximum clique in financial markets. For different thresholds we run the following procedure: we find the maximum clique and if its size is not greater than 2 we stop, otherwise we remove the stocks of this clique from the market and repeat the procedure. We have run this procedure on Russian and Swedish markets and compared the results with "stable" clustering. The comparison shows that all found cliques are always the subsets of our clusters (more than one clique can be in one cluster). For example, on Russian market the biggest cluster with 42 stocks of "stable" clustering includes two largest cliques of the size 17 and 5 and one clique of size 3 in a smaller cluster (see Fig. 1, where red nodes belong to the largest clique of size 17, blue nodes—to the clique of size 5, and yellow nodes—to the clique

Table 4 Stable number of clusters for different countries

Financial market	"Stable" number of clusters
USA	31
Russia	15
Sweden	12

Table 5 Medians for "stable" clustering on Swedish market

Stock's name	Company's name	Cluster's size
Bili-A.st	Bilia AB	11
Fag.st	Fagerhult, AB	4
Indu-C.st	IND.VAERDEN	39
Inve-B.st	Investor AB	64
Kinv-B.st	Investment AB Kinnevik	47
Mson-A.st	Midsona AB	5
Mtro-SDB-B.st	Metro International S.A.	3
NCC-B.st	NCC AB	22
Orti-B.st	Ortivus AB	3
SSAB-B.st	SSAB Swedish Steel AB	33
Svol-B.st	Svolder AB	11
Vnil-SDB.st	Vostok Nafta Investment Ltd.	24

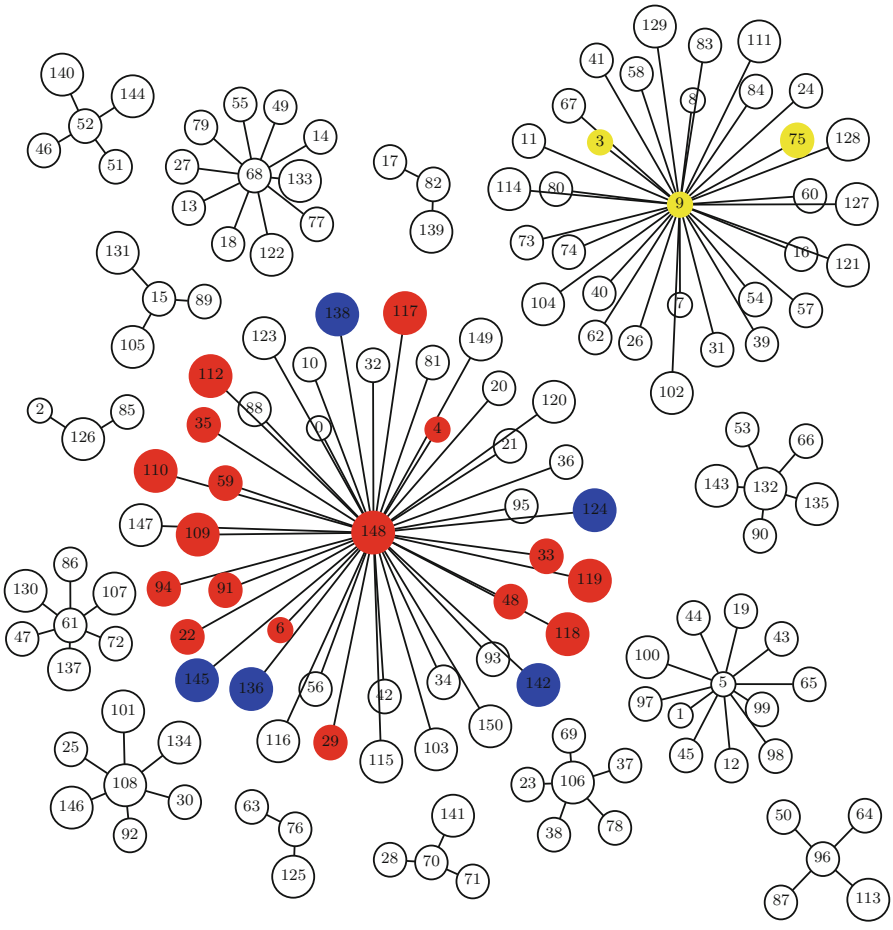


Fig. 1 Fifteen clusters and three cliques on Russian market

Table 6 The tightness in stars on Russian market divided into 15 stars

Cluster	Tightness	Cluster size	Cluster	Tightness	Cluster size
1	0.519449363	42	9	0.177086487	4
2	0.415963672	40	10	0.153309133	3
3	0.373704426	34	11	0.152456093	4
4	0.278391626	3	12	0.112344314	2
5	0.27505577	2	13	0	1
6	0.22470929	3	14	0	1
7	0.211888944	8	15	0	1
8	0.205771408	3			

of size 3). We believe that our approach gives more general information about the whole structure of the corresponding market since we use all correlation coefficients without any thresholds.

We have considered separately negative, positive, and absolute correlations for the markets. Our observations let us make a conclusion that negative correlations are not important for financial markets because the results for positive and absolute correlations are almost the same: for every number of clusters the set of medians are the same and the structure of clusters are almost the same. Thus we can remove negative correlations from the consideration or better substitute them by absolute values.

One more interesting result is that many medians in the market of the USA are the market indices and funds such as the Dow Jones, S&P, and others which form the largest clusters. But it is not surprising because such indices and funds are "linear combinations" of the stocks they consist of. Moreover, for the USA market the ordering of clusters by means of their tightness values in a non-increasing order is similar to the ordering of the same clusters by means of their cluster sizes (see Table 7). For the Russian market similar orderings are valid for the first three largest clusters only (see Table 6). Also the obtained largest clusters for the USA market are much more stable compared to the obtained largest clusters for the Russian market by means of the corresponding tightness values (see Tables 6 and 7).

7 Conclusions and Future Research Directions

In this paper we introduce a new method to analyze financial markets based on the PMP. We have implemented all recent discoveries within the pseudo-Boolean approach to solve the PMP presented in Goldengorin et al. [23] and have tested an exact and heuristic algorithms to solve the PMP. The approach we apply allows us to cluster a market into highly connected components in which stock prices are strongly correlated. We do not lose any financial information because we don't use any thresholds for correlation coefficients to construct a market graph. The method provides stable clustering of financial markets. Our approach outputs the set of p

Table 7 The tightness in stars on the USA market divided into 30 stars

Cluster	Tightness	Cluster size	Cluster	Tightness	Cluster size
1	0.993503666	412	16	0.367195974	96
2	0.985951948	368	17	0.359995482	78
3	0.91563294	309	18	0.338841811	73
4	0.89599758	294	19	0.280333928	68
5	0.875812686	186	20	0.247558649	61
6	0.781924916	143	21	0.222761999	54
7	0.743636731	125	22	0.18101556	46
8	0.725027741	123	23	0.179470838	33
9	0.668988825	120	24	0.170499094	31
10	0.656472908	119	25	0.161527103	28
11	0.515632467	109	26	0.157092122	27
12	0.515098615	107	27	0.086334098	27
13	0.485984984	107	28	0.054689215	12
14	0.44828545	105	29	0.045718395	11
15	0.39236848	100	30	0.023304706	7

clustered submarkets each of which might be represented by a leading stock, number of stocks within each cluster, and the average correlation coefficient T_i for all $i = 1, \ldots, p$.

The PMP-based approach can be used as an efficient aggregation tool for the further financial market analysis. We are able to substitute all stocks of the market by the found set p of stocks (called p medians in terms of the original PMP) because they still reflect the behavior of the whole market. So we can strongly reduce the size of the problem which is very useful for all computational methods applied to stock market analysis. Our representation of the whole stock market (with a huge number of stocks) by means of the p leading stocks might be very useful for traders as a tool to trade a large number of stocks simultaneously related to the chosen p leading stocks.

One of the most promising research directions is to apply a similar approach based on the Generalized PMP which is just the Simple Plant Location Problem with the fixed number p of opened sites (see AlBdaiwi et al. [20]). Another promising research direction is to use the bipartite and multidimensional matching of stocks (see Bekker et al. [24]) combined with the pre-selection of p-median stocks [25].

References

1. Mantegna, R.N.: Hierarchical structure in financial markets. Eur. Phys. J. B **11**, 193–197 (1999)
2. Kullmann, L., Kertesz, J., Mantegna, R.N.: Identification of clusters of companies in stock indices via Potts super-paramagnetic transactions. Physica A **287**, 412–419 (2000)
3. Onnela, J.-P., Chakraborti, A., Kaski, K., Kertesz, J.: Dynamic asset trees and portfolio analysis. Eur. Phys. J. B **30**, 285–288 (2002)

4. Onnela, J.-P., Chakraborti, A., Kaski, K., Kertesz, J., Kanto, A.: Asset trees and asset graphs in financial markets. Phys. Scr. **T106**, 48–54 (2003)
5. Cukur, S., Eryigit, M., Eryigit, R.: Cross correlations in an emerging market financial data. Physica A **376**, 555–564 (2007)
6. Onnela, J.-P., Chakraborti, A., Kaski, K., Kertesz, J.: Dynamic asset trees and Black Monday. Physica A **324**, 247–252 (2003)
7. Kenett, D.Y., Shapira, Y., Madi, A., Bransburg-Zabary, S., Gur-Gershgoren, G., Ben-Jacob, E.: Dynamics of stock market correlations. AUCO Czech Econ. Rev. **4**, 330–340 (2010)
8. Kenett, D.Y., Tumminello, M., Madi, A., Gur-Gershgoren, G., Mantegna, R.N.: Dominating clasp of the financial sector revealed by partial correlation analysis of the stock market. PLoS ONE **12**(5), 1–14 (2010)
9. Boginski, V., Butenko, S., Pardalos, P.M.: Statistical analysis of financial networks. Comput. Stat. Data Anal. **48**, 431–443 (2005)
10. Boginski, V., Butenko, S., Pardalos, P.M.: Mining market data: a network approach. Comput. Oper. Res. **33**, 3171–3184 (2006)
11. Jung, W.-S., Chae, S., Yang, J.-S., Moon, H.-T.: Characteristics of the Korean stock market correlations. Physica A **361**, 263–271 (2006)
12. Huang, W.-Q., Zhuang, X.-T., Yao, S.: A network analysis of the Chinese stock market. Physica A **388**, 2956–2964 (2009)
13. Tabak, B.M., Serra, T.R., Cajueiro, D.O.: Topological properties of stock market networks: the case of Brazil. Physica A **389**, 3240–3249 (2010)
14. Jallo, D., Budai, D., Boginski, V., Goldengorin, B., Pardalos, P.M.: Network-based representation of stock market dynamics: an application to American and Swedish stock markets. Springer Proc. Math. Stat. **32**, 91–108 (2013)
15. Reese, J.: Solution methods for the p-Median problem: an annotated bibliography. Networks **48**(3), 125–142 (2006)
16. Mladenovic, N., Brimberg, J., Hansen, P., Moreno-Perez, J.A.: The p-median problem: a survey of metaheuristic approaches. Eur. J. Oper. Res. **179**, 927–939 (2007)
17. Hammer, P.L.: Plant location - a pseudo-Boolean approach. Isr. J. Technol. **6**, 330–332 (1968)
18. Beresnev, V.L.: On a problem of mathematical standardization theory. Upravliajemyje Sistemy **11**, 43–54 (1973) [in Russian]
19. AlBdaiwi, B.F., Ghosh, D., Goldengorin, B.: Data aggregation for p-median problems. J. Comb. Optim. **3**(21), 348–363 (2011)
20. AlBdaiwi, B.F., Goldengorin, B., Sierksma, G.: Equivalent instances of the simple plant location problem. Comput. Math. Appl. **57**(5), 812–820 (2009)
21. Goldengorin, B., Krushinsky, D.: Complexity evaluation of benchmark instances for the p-median problem. Math. Comput. Model. **53**, 1719–1736 (2011)
22. Goldengorin, B., Krushinsky, D., Slomp, J.: Flexible PMP approach for large size cell formation. Oper. Res. **60**(5), 1157–1166 (2012)
23. Goldengorin, B., Krushinsky, D., Pardalos, P.M.: Cell Formation in Industrial Engineering. Theory, Algorithms and Experiments. Springer, Berlin, 218 pp. (2013). ISBN:978-1-4614-8001-3
24. Bekker, H., Braad, E.P., Goldengorin, B.: Using bipartite and multidimensional matching to select the roots of a system of polynomial equations. In: Computational Science and Its Applications—ICCSA. Lecture Notes in Computer Science, vol. 3483, pp. 397–406. Springer, Berlin (2005)
25. Goldengorin, B., Krushinsky, D.: A computational study of the pseudo-Boolean approach to the p-median problem applied to cell formation. In: Pahl, J., Reiners, T., Voß, S. (eds.) Network Optimization: Proceedings of 5th International Conference (INOC 2011), Hamburg, 13-16 June 2011. Lecture Notes in Computer Science, vol. 6701, pp. 503–516. Springer, Berlin (2011)

Clustering as an Approach to 3D Reconstruction Problem

Sergey Arkhangelskiy and Ilya Muchnik

Abstract Numerous applications of information technology are connected with 3D-reconstruction task. One of the important special cases is reconstruction using 3D point clouds that are collected by laser range finders and consumer devices like Microsoft Kinect. We present a novel procedure for 3D image registration that is a fundamental step in 3D objects reconstruction. This procedure reduces the task complexity by extracting small subset of potential matches which is enough for accurate registration. We obtain this subset as a result of clustering procedure applied to the broad set of potential matches, where the distance between matches reflects their consistency. Furthermore, we demonstrate the effectiveness of the proposed approach by a set of experiments in comparison with state-of-the-art techniques.

Keywords 3D object reconstruction • Cluster analysis applications • Point set registration

1 Introduction

3D object processing is an area of interest within a lot of applications, including robotics, engineering, medicine and entertainment. One of the most important tasks in this area is the reconstruction of 3D objects. Surface registration is the critical procedure in the process of reconstruction and it got a lot of research attention in recent years.

S. Arkhangelskiy (✉)
Moscow, Russia
e-mail: mc.vertix@gmail.com

I. Muchnik
DIMACS, Rutgers University, New Brunswick, NJ, USA
e-mail: muchnik@dimacs.rutgers.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_6,

Different devices based on laser scanners [1], time-of-flight sensors [2], or stereo vision [3] provide surface representations of different quality. The most common 3D object representation is a point cloud, that actually is just a set of three-dimensional points. Due to the limited field of view and occlusions the obtained clouds cover the object only partially. To reconstruct the entire object, a set of point clouds representing the object from different point of views is taken and then must be combined.

In this paper we address the problem of registering two point cloud inputs. The problem of registering several inputs may be reduced to the two-cloud problem by simple iterative procedure that registers inputs one by one.

There are three major groups of registration methods. First group represented by [4–6] looks at the problem in probabilistic setting. Each point cloud is represented as probability distribution, and then some distance between distributions is minimized. For instance, Tsin and Kanade [4] propose a distance measure proportional to the correlation of two density estimates. Jian and Vemuri [6] represent each cloud as Gaussian mixture and align clouds minimizing L_2 distance between distributions. This approach is robust to outliers and can be organically modified to nonrigid transformations case.

Second and the third group use notion of point correspondences and obtain the transformation between clouds as the best alignment for the matched points. Fine registration algorithms are mostly represented by descendants of Iterative Closest Points (ICP) method, originally proposed by Chen and Medioni [7] and Besl and McKay [8]. These methods iteratively improve the registration by minimizing the distance between pairs of selected matching points. Thus the quality of final registration strongly depends on initial alignment.

The third group—coarse registration methods include a group of algorithms that leverage from local surface descriptors which are intrinsic to the shape and do not depend on the initial surfaces alignment. The popular examples of such descriptors are Point Signatures [9] and Spin Images [10]. Other coarse registration algorithms like RANSAC-based DARCES [11] and congruent 4-point sets [12] exploit different ideas.

Most of both fine and coarse registration methods are two-step. On the first step they detect correspondences between surface points. For example, ICP-like methods establish matches by finding for each point of one cloud the closest point on another cloud [8]. Descriptor-based methods perform search in descriptor space and detect pairs of similar points.

On the second step these algorithms find a rigid transformation that aligns correspondent points. Most methods obtain it as a solution of the least square minimization problem using any of existing techniques [13, 14]. In order to get a meaningful solution, the correspondences have to be geometrically consistent. Otherwise one or several outliers present in the data can significantly affect the final registration.

The problem of extracting the consistent matches subset is not new in 3D shape registration. Known methods include adapted random sample consensus (RANSAC) [15–17], genetic algorithms [18, 19], and usage of pairwise correspondence consistency measures [20].

Recently Albarelli et al. suggested another interesting approach for filtering correspondences [21]. They cast the selection problem in a game-theoretic framework, where mating points that satisfy mutual consistency constraint thrive, eliminating all other correspondences.

In this paper we propose another method for extracting the globally consistent matches set. Our method is inspired by the ideas of clustering analysis. The rest of this paper is organized as follows. In Sect. 2 we define what we call matches graph, designed to represent the information about geometric consistency between correspondences. We show how the problem of extracting globally consistent matches subset is connected with the problem of extracting sparse graph core, that is known in clustering analysis. Then, in Sect. 3 we propose a variant of Layered clusters [22, 23] technique that we apply to our graph. In Sect. 4 we use this technique to solve the registration problem and provide an experimental validation of our suggestions, including comparison with other state-of-the art methods.

2 Matches Graph

Let $C = \{x_1, x_2, \ldots, x_{|C|}\}$ be the first point cloud, and $D = \{y_1, y_2, \ldots, y_{|D|}\}$ be the second point cloud. They both represent parts of the same object. If these two parts overlap, there are subclouds C' and D' that are images of the overlapping region. Every point on object surface within the overlapping region has images $x \in C'$ and $y \in D'$ that match each other. The rigid transformation T that aligns C' with D' coincides the mating points, i.e. $Tx \approx y$. Non exact equality is the result of noise and discretization in object measurements.

As we said in previous section, the set of correct matches $\{[x_i, y_i]\}$ gives us a way to find the registration T by solving the following optimization problem [14]:

$$\sum_i ||Tx_i - y_i||^2 \to \min \tag{1}$$

Henceforth our goal is to find this subset.

There are $|C| \times |D|$ possible matches, but only small share of them is correct. Following many other coarse registration methods [9, 10, 24], we use surface descriptor matching to reduce the number of considered putative correspondences. We chose Fast Point Feature Histograms (FPFH) feature proposed by Rusu et al. [17] as a descriptor. FPFH $Q(x)$ of a point x is a 33-dimensional real-valued vector that is intrinsic to the shape of point vicinity and is invariant to viewpoint changes. It is designed to be stable to noise and surface discretization errors.

For every point $x \in C$ we compute its descriptor $Q(x)$ and find k points from D with the closest descriptors (k—is external parameter). Thereby we get k possible matches for one point, and $k \times |C|$ in total. When we take k closest points from C for every point in D, we get another $k \times |D|$ matches. The intersection M of these two sets contains not more than $k \times \min(|C|, |D|)$ assumed correspondences.

Though due to the usage of descriptors M contains only matches with locally similar points, some of these matches can still be wrong. In order to extract correct matches we need to exploit some other information despite the local surface similarity.

Geometric consistency of matches is exactly this type of information. Consider two matches $[x_i, y_i]$ and $[x_j, y_j]$ and registration T ($Tx_i = y_i, Tx_j = y_j$). As far as T is a rigid transformation, we have:

$$\big|\big|x_i - x_j\big|\big| = \big|\big|y_i - y_j\big|\big| \tag{2}$$

More generally, the following inequality holds:

$$\begin{aligned} &||Tx_i - y_i|| + \big|\big|Tx_j - y_j\big|\big| \\ &\quad \geq \big|\big|Tx_i - y_i + y_j - Tx_j\big|\big| = \big|\big|T(x_i - x_j) - (y_i - y_j)\big|\big| \\ &\quad \geq \big|\,\big|\big|x_i - x_j\big|\big| - \big|\big|y_i - y_j\big|\big|\,\big| \end{aligned}$$

It means that for every pair of correct matches $\big|\big|x_i - x_j\big|\big| \approx \big|\big|y_i - y_j\big|\big|$.

We call continuous function $\varphi(x_i, y_i, x_j, y_j)$ a *geometric consistency function*, if it satisfies the following properties.

1. Symmetry:

$$\varphi(x_i, y_i, x_j, y_j) = \varphi(x_j, y_j, x_i, y_i) \tag{3}$$

2. Nonnegativeness

$$\varphi(x_i, y_i, x_j, y_j) \geq 0 \tag{4}$$

3. Criteria of equality to zero:

$$\varphi = 0 \text{ only if } \big|\big|x_i - x_j\big|\big| = \big|\big|y_i - y_j\big|\big| \tag{5}$$

4. Monotonicity.

$$\begin{aligned} &\varphi(x_0, y_0, x_1, y_1) \leq \varphi(x_0, y_0, x_2, y_2) \text{ if and only if} \\ &\big|\,||x_0 - x_1|| - ||y_0 - y_1||\,\big| \leq \big|\,||x_0 - x_2|| - ||y_0 - y_2||\,\big| \end{aligned} \tag{6}$$

These functions reflect the notion of consistency, i.e., the matches are more consistent in some sense if function value is close to 0, and less consistent if it is big.

The following functions are examples of consistency functions:

$$\varphi_0(x_i, y_i, x_j, y_j) = \left| \left|\left|x_i - x_j\right|\right| - \left|\left|y_i - y_j\right|\right| \right|$$

$$\varphi_1(x_i, y_i, x_j, y_j) = \frac{\left| \left|\left|x_i - x_j\right|\right| - \left|\left|y_i - y_j\right|\right| \right|}{\max(\left|\left|x_i - x_j\right|\right|, \left|\left|y_i - y_j\right|\right|)}$$

$$\varphi_2(x_i, y_i, x_j, y_j) = \frac{\left| \left|\left|x_i - x_j\right|\right| - \left|\left|y_i - y_j\right|\right| \right|}{\frac{1}{2}(\left|\left|x_i - x_j\right|\right| + \left|\left|y_i - y_j\right|\right|)}$$

Function φ_0 may be called absolute consistency function, as its value represents the absolute difference between distances. In their turn, φ_1 and φ_2 are the functions of relative consistency functions.

In our experiments we use function φ_1 that we can also be rewritten as:

$$\varphi_1(x_i, y_i, x_j, y_j) = 1 - \min\left[\frac{\left|\left|x_i - x_j\right|\right|}{\left|\left|y_i - y_j\right|\right|}, \frac{\left|\left|y_i - y_j\right|\right|}{\left|\left|x_i - x_j\right|\right|}\right] \tag{7}$$

Now we introduce the matches graph G, which nodes are the matches from M and that has every pair of nodes connected with an edge. The chosen geometric consistency function φ defines the weights of edges connecting putative correspondences

$$w_{[x_i, y_i], [x_j, y_j]} = \varphi(x_i, y_i, x_j, y_j).$$

3 Extraction of Graph Sparse Core

As we discussed earlier, every correct correspondences pair is geometrically consistent. In terms of our graph we may say that the subset of correct matches is sparse, i.e., weights of edges connecting the nodes within this subset are small.

This observation refers us to the problem of finding the dense (sparse) components in a graph [25, 26]. In this paper we use layered clusters technique, developed in the works of Mirkin et al. [22, 23] to approach this problem.

3.1 Layered Clusters

The problem of finding the dense (sparse) core of a finite set I is formalized in terms of a set function $F(H)$, where $H \subseteq I$.

Let us consider the so-called tightness set functions. These functions are defined using linkage function $\pi(\alpha, H)$, where $\alpha \in H$, $H \subseteq I$. The linkage function represents how strongly element i is connected with other elements of set H. We say that function $\pi(\alpha, H)$ is monotone if for any $H_1 \subseteq H_2 \subseteq I$ $\pi(\alpha, H_1) \leq \pi(\alpha, H_2)$. In our work we use function

$$\pi(\alpha, H) = \sum_{\beta \in H} \varphi(x_\alpha, y_\alpha, x_\beta, y_\beta) \tag{8}$$

though other functions are possible [22].

Function $F(H)$ defined as

$$F(H) = \max_{\alpha \in H} \pi(\alpha, H) \tag{9}$$

naturally produces the subset sequence $I = H_0 \supset H_1 \supset \ldots \supset H_N = \emptyset$, called layered clusters [23]. In this sequence, each next subset comes out of the previous one by subtracting one element. This is the element that has the biggest linkage value among elements of H_i:

$$\alpha_i = \max_{\alpha \in H_{i-1}} \pi(\alpha, H_{i-1}), \tag{10}$$

$$H_i = H_{i-1} \setminus \{\alpha_i\}. \tag{11}$$

For every set H_i we can compute average distance between points within this subset:

$$\Phi(H_i) = \frac{2}{|H_i| \cdot (|H_i| - 1)} \sum_{\beta, \gamma \in H_i} \varphi(x_\beta, y_\beta, x_\gamma, y_\gamma) \tag{12}$$

This value reflects how "tight" this subset is. If this value is low, when the matches within dataset are consistent and "in average," don't conflict with each other.

Our algorithm extracts the biggest subset H_{i*} that has average distance less than given threshold τ as a sparse core of our graph, and consequently as a set of correct matches. We solve problem (1) using the obtained matches to compute the registration T.

4 Experimental Results

We have conducted our experiments on model "Armadillo" [27] from Stanford 3D Repository (Fig. 1). We took one partial point cloud of the model and made further experiments with it.

Fig. 1 Two scans of Armadillo registered with our method

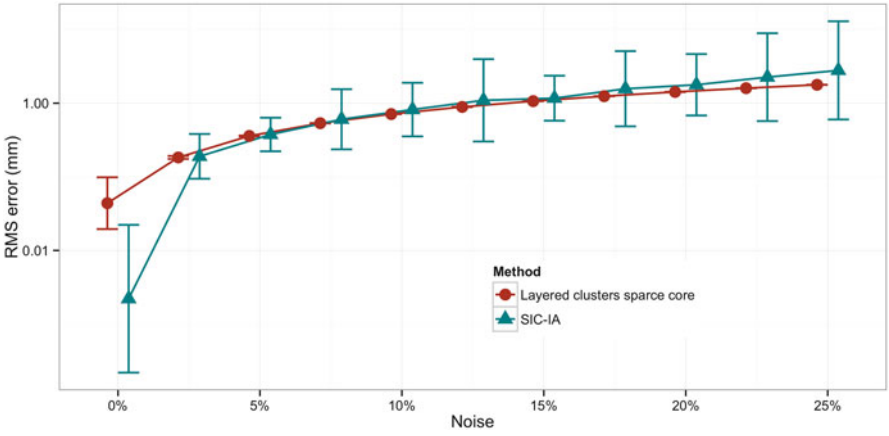


Fig. 2 Comparison of our and SAC-IA methods, measuring RMS error as a function of noise

4.1 Sensitivity to Noise

In the first set of experiments we distorted point cloud with Gaussian noise and randomly changed coordinate frame. Rotation is specified by uniformly sampled unit quaternion, and translation vector has Gaussian distribution with zero mean and unit covariance matrix. The noise level is defined as the ratio of the noise standard deviation and the median distance between neighbor points in the cloud. After that, we registered the original cloud to the distorted one and measured the root mean square (RMS) error and the estimation errors of the translation vector and rotation angle.

In Fig. 2 we compare the performance of our method and sample consensus initial alignment (SAC-IA) method presented in [17] and implemented in Point Cloud Library [28].

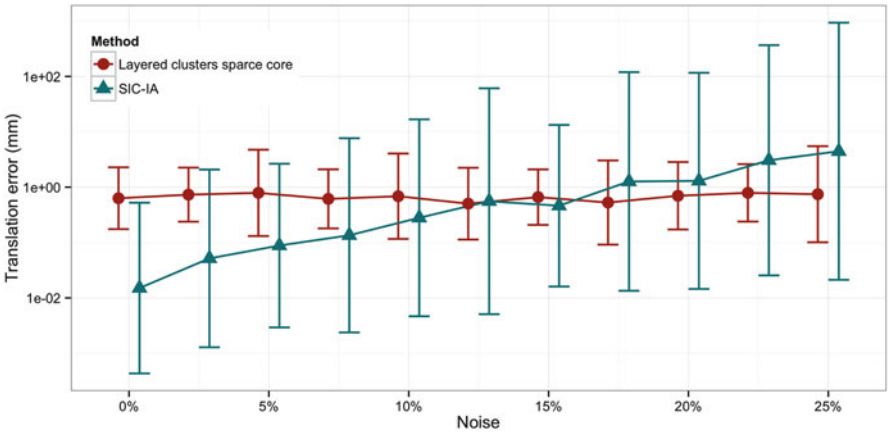


Fig. 3 Comparison of our method and RANSAC-method, measuring translation error as a function of noise

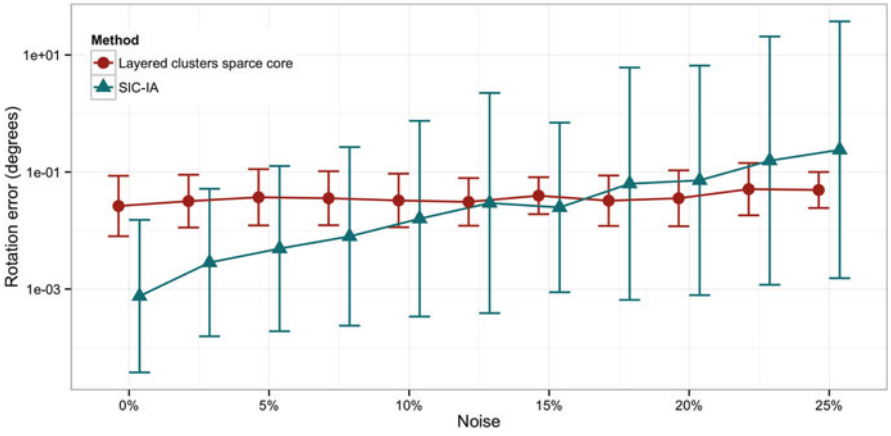


Fig. 4 Comparison of our method and RANSAC-method, measuring rotation error as a function of noise

In this method we randomly take 3-element subsets from our original matches set. Using the least-squares method, for each subset we determine the transformation that aligns chosen three pairs of points. For every transformation we compute the number of remaining matches that have registration residual below some threshold. The transformation with the biggest number of such matches is taken as the registering transformation.

In Figs. 3 and 4 we show the estimation errors of translation vector and rotation angle for different methods as a function of noise level. In the experiment reflected in Figs. 2, 3 and 4 we use parameter $k = 2$ and $\tau = 0.01$ (1 %). We may see that our method is more stable than the reference method starting from 2.5 % noise level.

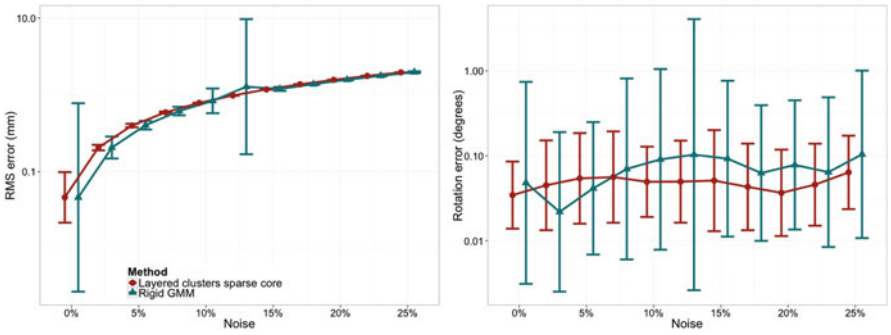


Fig. 5 Comparison of our method and Gaussian Mixture Models method, measuring RMS and rotation error as a function of noise

In the next experiment, we compared our method with the state-of-the-art Gaussian Mixture Models method [6] using author's implementation.[1] We used default sequence of scale parameters equal to 0.5, 0.1, and 0.02 of model size, where model size was estimated as RMS distance from model points to their centroid. This method is sensitive to the initial alignment, and for big rotation angles it often converged to wrong alignments. We limited rotation angle by 72° and didn't apply translation to coordinate frame. We also downsampled the cloud from 32,300 to about 6,500 points.

For every noise level we made 15 alignment experiments with different rotations, and measured RMS and rotation estimation errors. The results are presented in Fig. 5. The big confidence interval of GMM method on 12.5 % noise level is explained by experiment run where GMM method was trapped in local minima and rotation error reached 36°. Our method is robust to initial alignment and has comparable performance.

4.2 Sensitivity to Outliers Presence

In order to estimate robustness of proposed method to the presence of outliers, we extended two copies of the model with set of points uniformly generated within bounding box of the model. Each copy was extended with different set of outliers, and the number of points was changing from 10 to 100 % of original model size. Similarly to the previous experiment one of the copies was randomly transformed to another coordinate frame, but in GMM experiment no translation was applied and rotation angle was limited.

[1] https://code.google.com/p/gmmreg.

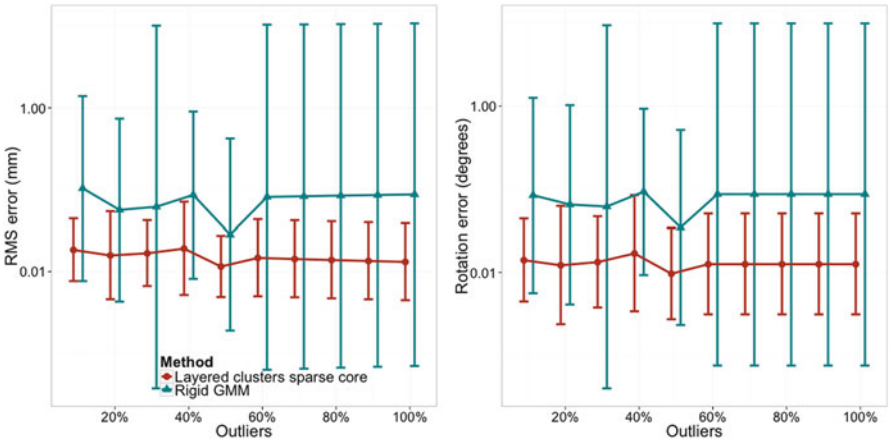


Fig. 6 Comparison of our method and Gaussian Mixture Models method, measuring RMS and rotation error as a function of outliers number

For given amount of outliers each method was tested in 15 experiments, with different transform and outliers on each run. Both methods, ours and GMM, are pretty accurate and not sensitive to outlier presence with our method being slightly better (see Fig. 6).

5 Conclusion

In this paper we have presented a novel approach to the point cloud registration problem that casts it to the clustering framework. This brings the power of clustering techniques to the field of three-dimensional reconstruction.

For instance, we applied layered clusters techniques to the matches graph and demonstrated that the set of inliers detected by our method suffices for the high quality registration. Our experiments show that the suggested method performance is on par with an industry state-of-the-art method.

In the future we plan to adapt other clustering methods like K-means and spectral clustering to the matches graph. We are also working on extending our algorithm to the case of simultaneous registration of multiple point clouds.

References

1. Forest, J., Salvi, J.: An overview of laser slit 3d digitasers. In: International Conference on Robots and Systems, vol. 1, pp. 73–78 (2002)
2. Ullrich, A., Studnicka, N., Riegl, J., Orlandini, S.: Long-range high-performance time-of-flight-based 3D imaging sensors. In: Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission, 2002, pp. 852–855. IEEE, New York (2002)

3. Matabosch, C., Salvi, J., Forest, J.: Stereo rig geometry determination by fundamental matrix decomposition. In: Workshop on European Scientific and Industrial Collaboration, vol. 2, pp. 405–412 (2003)
4. Tsin, Y., Kanade, T.: A correlation-based approach to robust point set registration. In: Computer Vision-ECCV 2004, pp. 558–569. Springer, Berlin (2004)
5. Myronenko, A., Song, X.S.X.: Point set registration: coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. **32**(12), 2262–2275 (2010)
6. Jian, B., Vemuri, B.C.: Robust point set registration using gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 33(8), 1633–1645 (2011)
7. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: Proceedings 1991 IEEE International Conference on Robotics and Automation, pp. 2724–2729. IEEE Computer Society Press, Los Alamitos (1991)
8. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
9. Chua, C.S., Jarvis, R.: Point signatures: a new representation for 3D object recognition. Int. J. Comput. Vis. **25**(1), 63–85 (1997)
10. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. **21**(5), 433–449 (1999)
11. Chen, C.S., Hung, Y.P., Cheng, J.B.: RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. IEEE Trans. Pattern Anal. Mach. Intell. **21**(11), 1229–1234 (1999)
12. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-Points congruent sets for robust pairwise surface registration. ACM Trans. Graph. **27**(3), 1 (2008)
13. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. Opt. Soc. Am. **4**, 629–642 (1987)
14. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intell. **9**(5), 698–700 (1987)
15. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
16. Feldmar, J., Ayache, N.: Rigid, affine and locally affine registration of free-form surfaces. Int. J. Comput. Vis. **18**(2), 99–119 (1996)
17. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D registration. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3212–3217 (2009)
18. Brunnstrom, K., Stoddart, A.J.: Genetic algorithms for free-form surface matching. In: Proceedings of the 13th International Conference on Pattern Recognition, August 1996, vol. 4, pp. 689–693. IEEE, New York (1996)
19. Chow, C.K., Tsui, H.T., Lee, T.: Surface registration using a dynamic genetic algorithm. Pattern Recognit. **37**(1), 105–117 (2004)
20. Johnson, A.E., Hebert, M.: Surface matching for object recognition in complex three-dimensional scenes. Image Vis. Comput. **16**(9–10), 635–651 (1998)
21. Albarelli, A., Rodola, E., Torsello, A.: A game-theoretic approach to fine surface registration without initial motion estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2010, pp. 430–437. IEEE, New York (2010)
22. Mirkin, B., Muchnik, I.: Combinatoral optimization in clustering. In: Handbook of Combinatorial Optimization, pp. 1007–1075. Springer, New York (1999)
23. Mirkin, B., Muchnik, I.: Layered clusters of tightness set functions. Appl. Math. Lett. **15**(2), 147–151 (2002)
24. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Computer Vision—ECCV 2010, pp. 356–369 (2010)
25. Charikar, M.: Greedy approximation algorithms for finding dense components in a graph. In: Approximation Algorithms for Combinatorial Optimization, pp. 84–95 (2000)

26. Le, T.V., Kulikowski, C.A., Muchnik, I.B.: Coring method for clustering a graph. In: 19th International Conference on Pattern Recognition (ICPR 2008), December 2008, pp. 1–4. IEEE, New York (2008)
27. Krishnamurthy, V., Levoy, M.: Fitting smooth surfaces to dense polygon meshes. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '96, pp. 313–324. ACM, New York (1996)
28. Rusu, R.B.: Point Cloud Library. Willow Garage, Menlo Park (2010)

Selecting the Minkowski Exponent for Intelligent K-Means with Feature Weighting

Renato Cordeiro de Amorim and Boris Mirkin

Abstract Recently, a three-stage version of K-Means has been introduced, at which not only clusters and their centers, but also feature weights are adjusted to minimize the summary p-th power of the Minkowski p-distance between entities and centroids of their clusters. The value of the Minkowski exponent p appears to be instrumental in the ability of the method to recover clusters hidden in data. This paper advances into the problem of finding the best p for a Minkowski metric-based version of K-Means, in each of the following two settings: semi-supervised and unsupervised. This paper presents experimental evidence that solutions found with the proposed approaches are sufficiently close to the optimum.

Keywords Clustering • Minkowski metric • Feature weighting • K-Means

1 Motivation and Background

Clustering is one of the key tools in data analysis. It is used particularly in the creation of taxonomies when there are no accurate labels available identifying any taxon, or not enough such labels to train a supervised algorithm. K-Means is arguably the most popular clustering algorithm being actively used by practitioners in data mining, marketing research, gene expression analysis, etc. For example, a

R.C. de Amorim (✉)
Department of Computing, Glyndŵr University, Wrexham LL11 2AW, UK
e-mail: r.amorim@glyndwr.ac.uk

B. Mirkin
Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, Moscow, Russian Federation

Department of Computer Science, Birkbeck University of London, United Kingdom
e-mail: bmirkin@hse.ru; mirkin@dcs.bbk.ac.uk

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_7,

Google search made on the 15th of March 2013 returned 473 mln pages to the query "k-means" and only 198 mln pages to the query "cluster," despite the latter being more general. Thanks to its popularity, K-Means can be found in a number of software packages used in data analysis, including MATLAB, R and SPSS.

K-Means aims to partition a dataset represented by a matrix $Y = (y_{iv})$, where y_{iv} represents the value of feature v, $v = 1, \ldots, V$, on entity $i \in I$, into K homogeneous clusters S_k, together with their centroids $c_k (k = 1, 2, \ldots, K)$. Given a measure of distance $d(y_i, c_k)$, where y_i denotes i-th row $y_i = (y_{i1}, y_{i2}, \ldots, y_{iV})$ of Y, K-Means iteratively minimizes the summary distance between the entities y_i and centroids c_k of their clusters

$$W(S, C) = \sum_{k=1}^{K} \sum_{i \in S_k} d(y_i, c_k) \tag{1}$$

The minimization process works according to the alternating optimization scheme. Given the centroids c_k, the optimal clusters S_k are found to minimize criterion (1) over clusters. Given the found clusters S_k, the optimal centroids c_k are found by minimizing criterion (1) over centroids. This is especially simple when the scoring function $d(y_i, c_k)$ is the squared Euclidean distance $d(y_i, c_k) = \sum_v (y_{iv} - c_{kv})^2$. In this case, the optimal centroids are the clusters means, and the optimal clusters consist of entities that are nearest to their centroids, clearly favoring spherical clusters. The iterations stop when the centroids stabilize; the convergence is warranted by the fact that the criterion decreases at every step, whereas the number of possible partitions is finite.

Being much intuitive and simple computationally, K-Means is known to have a number of drawbacks; among them are the following: (1) the method requires the number of clusters to be known beforehand; (2) the final clustering is highly dependent on the initial centroids it is fed; (3) the results highly depend on feature scaling.

Building on the work by Makarenkov and Legendre [11], Huang et al. [3, 9, 10], Mirkin [12], and Chiang and Mirkin [4], Amorim and Mirkin have introduced what they call the intelligent Minkowski Weighted K-Means (iMWK-Means) algorithm which mitigates the mentioned drawbacks [7]. This approach extends the K-Means criterion by distinguishing in it the feature weighting component, while using the Minkowski metric and initializing the process with anomalous clusters.

The superiority of iMWK-Means in relation to other feature-weight maintaining algorithms was experimentally demonstrated on medium-sized datasets [6, 7] including a record minimum number of misclassified entities, 5, on the celebrated Iris dataset [2]. Yet choosing the "right" exponent remains an issue. This has been addressed by Huang et al. [3, 9, 10] (weight exponent), Amorim and Mirkin [7] (Minkowski exponent) by exploring the similarity between the found clustering and the "right" partition on a range of possible values of the exponent and choosing the exponent value corresponding to the best match between the clustering and the partition.

In both cases the accuracy of cluster recovery appears to be highly dependent on the exponent value, which may drastically differ at different datasets. Yet in real-world problems the clusters in data are not known beforehand so this approach would not work. There can be two types of real-world scenarios:

1. semi-supervised clustering in which right cluster labels can be supplied for a small part of the data before the process of clustering;
2. unsupervised clustering in which no cluster labels are supplied beforehand at all.

This paper addresses the problem of choosing the Minkowski exponent in both scenarios. We experimentally investigate how our semi-supervised algorithm in scenario (1) works at different proportions of labelled data. We empirically demonstrate that it is possible to recover a good Minkowski exponent with as low as 5 % of data being labelled, and that large increases in this proportion of labelled data tend to have a small effect on cluster recovery. In scenario (2), we look at applying various characteristics of the cluster structure as potential indexes for choosing the right Minkowski exponent. Among them we introduce an index based on the iMWK-Means criterion and, also, indexes related to the so-called silhouette width [13]. It appears our approaches show rather satisfactory results.

The remainder is structured as follows. Section 2 describes the generic iMWK-Means in greater detail. Section 3 describes adaptations of iMWK-Means to the semi-supervised and unsupervised clustering situations. Section 4 presents our experimental setting, with both real-world benchmark data and synthetic datasets, and the experimental results. Section 5 concludes the paper.

2 Minkowski Weighted K-Means and iMWK-Means

Weighted K-Means (WK-Means) automatically calculates the weight of each feature conforming to the intuitive idea that features with low within-cluster variances are more relevant for the clustering than those with high within-cluster variances. Each weight can be computed both for the entire dataset, w_v, or within-clusters, w_{kv}. We utilize the latter approach involving cluster-specific feature weights. The WK-Means criterion by Huang et al. [3] is as follows:

$$W(S, C, w) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} w_{kv}^{p} |y_{iv} - c_{kv}|^2 \tag{2}$$

where V is the number of features in Y; $w = (w_{kv})$, the set of non-negative within-cluster feature weights such that $\sum_{v=1}^{V} w_{kv} = 1$ for each $k = 1, 2, \ldots, K$; and p, the adjustable weight exponent. This criterion is subjective to a crisp clustering, in which a given entity y_i can only be assigned to a single cluster. The MWK-Means [7] is a further extension of the criterion, in which the squared Euclidean distance

is changed for the p-th power of the Minkowski p-distance. The Minkowski p-distance between a given entity y_i and centroid c_k is defined below:

$$d_p(y_i, c_k) = \left(\sum_{v=1}^{V} |y_{iv} - c_{kv}|^p \right)^{1/p} \tag{3}$$

By separating positive measurement scales w_{kv} of features v in the coordinates of y_i and c_k, the p-th power of the Minkowski p-distance, hence without the $1/p$ exponent, can be expressed as:

$$d_{wp}(y_i, c_k) = \sum_{v=1}^{V} w_{kv}^p |y_{iv} - c_{kv}|^p \tag{4}$$

Putting (4) into criterion (2), one arrives at the Minkowski Weighted K-Means criterion:

$$W(S, C, w) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} w_{kv}^p |y_{iv} - c_{kv}|^p \tag{5}$$

Because of the additivity of the criterion, the weights can be set to be cluster-specific, so that any feature v may have different weights at different clusters k as reflected in (5). Given the partition and centroids; the weights are computed according to equations derived from the first-order optimality condition for criterion (5):

$$w_{kv} = \frac{1}{\sum_{u \in V} [D_{kv}/D_{ku}]^{1/(p-1)}} \tag{6}$$

where $D_{kv} = \sum_{i \in S_k} |y_{iv} - c_{kv}|^p$ and k is an arbitrary cluster [7]. In our experiments we have added a very small constant to the dispersions, avoiding any issue related to a dispersion being equal to zero. Equation (6) at $p = 1$ may seem problematic at first; however, such case is in fact of simple resolution. At $p = 1$, Eq. (6) is equivalent to a weight of one for the feature v with the smallest dispersion, and weights of zero in the others, all cluster-specific [3, 7]. The MWK-Means iterative minimization of criterion (5) is similar to the K-Means algorithm, but each iteration here consists of three stages, rather than the two stages of the generic K-Means, to take the computation of weights into account.

MWK-Means algorithm

1. Initialization. Define a value for the Minkowski exponent p. Select K centroids from the dataset at random. Set $v_{ik} = 1/V$.
2. Cluster update. Assign each entity to its closest centroid applying the Minkowski weighted distance (4).

3. Centroid update. Calculate the cluster centroids as the within-cluster Minkowski centers. Should the centroids remain unchanged, stop the process and output the results.
4. Weight update. Given clusters and centroids, compute the feature weights using Eq. (6). Go back to Step 2 for the next iteration.

The original K-Means algorithm makes use of the squared Euclidean distance, which favors spherical clusters. The MWK-Means criterion (5) favors any interpolation between diamond and square shapes, depending on the value of p. This property of MWK-Means makes the selection of p rather important for cluster recovery.

The Minkowski centers can be computed using a steepest descent algorithm from Amorim and Mirkin [7]. More precisely, given a series of reals, $y_1, \ldots, y_n$, its Minkowski p-center is defined as c minimizing the summary value

$$d(c) = \sum_{i=1}^{n} |y_i - c|^p \tag{7}$$

The algorithm is based on the property that $d(c)$ in (7) is convex for $p > 1$, and it uses the first derivative of $d(c)$ equal to $'d(c) = p(\sum_{i \in I+}(c - y_i)^{p-1} - \sum_{i \in I-}(y_i - c)^{p-1}$, where I^+ is the set of indices i at which $c > y_i$, and I^- is the set of indices i at which $c \leq y_i, i = 1, \ldots, n$.

Minkowski center algorithm

1. Sort given reals in the ascending order so that $y_1 \leq y_2 \leq \ldots \leq y_n$.
2. Initialize with $c_0 = y_{i*}$, the minimizer of $d(c)$ on the set y_i and a positive learning rate λ that can be taken, say, as 10 % of the range $y_n - y_1$.
3. Compute $c_0 - \lambda d'(c_0)$ and take it as c_1 if it falls within the minimal interval $(y_{i'}, y_{i''})$ containing y_{i*} and such that $d(y_{i'}) > d(y_{i*}), d(y_{i''}) > d(y_{i*})$. Otherwise, decrease λ a bit, say, by 10 %, and repeat the step.
4. Test whether c_1 and c_0 coincide up to a pre-specified precision threshold. If yes, halt the process and output c_1 as the optimal value of c. If not, move on.
5. Test whether $d(c_1) \leq d(c_0)$. If yes, set $c_0 = c_1$ and $d(c_0) = d(c_1)$, and go to step 2. If not, decrease λ a bit, say by 10 %, and go to step 3 without changing c_0.

Similarly to K-Means, the MWK-Means results highly depend on the choice of the initial centroids. This problem has been addressed for K-Means by using initialization algorithms that provide K-Means with a set of good centroids. Taking this into account, we have adapted the Anomalous Pattern algorithm from Mirkin [12] to supply MWK-Means with initial centroids. A combination of K-Means with the preceding Anomalous Pattern algorithm is referred to as the intelligent K-Means, iK-Means, in Mirkin [12]. The iK-Means has proved superior in cluster recovery over several popular criteria in experiments reported by Chiang and Mirkin [4]. The modified Anomalous Pattern algorithm involves the weighted Minkowski metric (4) with the weights computed according to Eq. (6). Together with the Anomalous Pattern initialization, MWK-Means forms what is referred to as the

intelligent Minkowski Weighted K-Means algorithm (iMWK-Means), ceasing to be a non-deterministic algorithm. Its formulation is as follows.

iMWK-Means algorithm

1. Sort all the entities according to their Minkowski weighted distance (4) to the Minkowski center of the dataset, c_c, using the Minkowski weighted metric and $1/V$ for each weight.
2. Select the farthest entity from the Minkowski center, c_t, as a tentative anomalous centroid.
3. Assign each of the entities to its nearest centroid of the pair, c_c and c_t, according to the Minkowski weighted metric.
4. Compute c_t as the Minkowski center of its cluster.
5. Update the weights according to Eq. (6). If c_t has moved on step 4, return to step 3 for the next iteration.
6. Set all the entities assigned to c_t as an anomalous cluster and remove it from the dataset. If there are still unclustered entities remaining in the dataset, return to step 2 to find the next anomalous cluster.
7. Run MWK-Means using the centroids of the K anomalous clusters with the largest cardinality.

3 Selection of the Minkowski Exponent in the Semi- and Un-supervised Settings

In this section we present methods for selecting the Minkowski exponent in each of the two settings, semi-supervised and unsupervised. The first utilizes a small portion of data that have been labelled; the second, a cluster-scoring function over partitions.

3.1 Choosing the Minkowski Exponent in the Semi-supervised Setting

The semi-supervised setting relates to the scenario in which the cluster labels are known not for all, but only for a relatively small proportion q of the dataset being clustered. Then either of two options can be taken: (a) first, cluster only those labelled entities to learn the best value for the Minkowski exponent p as that leading to the partition best matching the pre-specified labels, then using the learnt p, cluster the entire dataset, or (b) to cluster all the entities and learn the best p at the labelled part of the dataset. In Amorim and Mirkin [7] the option (a) has been disapproved rather convincingly. Therefore, only option (b) is tested in this paper at differing values of q.

The algorithm comprising both learning and testing p runs at a dataset, at which all the pre-specified clustering labels are known, as follows:

Run R = 50 times:

1. Get a random sample of labelled entities of size q: cluster labels on this set are assumed to be known.
2. Run iMWK-Means over the whole dataset with p taken in the interval from 1 to 5 in steps of 0.1.
3. Select the Minkowski exponent with the highest accuracy achieved on the entities whose labels are known as p^*.
4. Calculate the accuracy using p^* and the whole dataset by comparing the found partition and that one pre-specified.

Calculate the average accuracy of the R runs and standard deviation.

Our search for a good p occurs in the interval $[1, 5]$. We have chosen the lower bound of one because this is the minimum for which we can calculate a Minkowski center (median), since Eq. (7) is convex for $p > 1$. We have chosen the upper bound of five following our previous experiments [7].

Of course, in a real-life clustering scenario one would not normally have access to the labels of the whole dataset. Here we utilize it only for evaluation purposes.

3.2 *Choosing the Minkowski Exponent in an Unsupervised Setting*

When no prior information of the hidden partition is available, a reasonable idea would be to find such a scoring function over the iMWK-Means partitions, that reaches its extreme value, that is, the maximum or minimum, at a resulting partition that is most similar to the hidden partition.

Under the original K-Means framework, one of the most popular scoring functions is the output of the K-Means criterion itself. One simply runs K-Means a number of times and sets the optimal partition to be that with the smallest sum of distances between the entities and their respective centroids. Unfortunately the iMWK-Means criterion (5) is not comparable at different ps, making its raw value inappropriate for a scoring function. However, we can normalize (5) in such a way that its dependence on p can be disregarded, we called it the Minkowski clustering index (MCI). For comparison, we also experiment with the so-called silhouette width [13] which has been reported as a good index in various empirical studies, one of the latest being by Arbelaitz et al. [1].

Let us define MCI, an index based on the minimized value of the MWK-Means criterion. We normalize the criterion over what may be called the Minkowski data p-scatter according to the feature weights found as an output of a run of the iMWK-Means:

$$\mathrm{MCI} = \frac{W_p(S, C, w)}{\sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} |w_{kv} y_{iv}|^p} \tag{8}$$

This index is comparable at clusterings with different ps. We choose that p at which the MCI is at its minimum.

Let us turn now to the silhouette width-based indexes. Given a set of clusters and an entity-to-entity dissimilarity measure, the silhouette width of an entity is defined as the relative difference between the average dissimilarity of that entity from the other entities in its cluster compared to its average dissimilarities to other clusters according to formula:

$$S(y_i) = \frac{b(y_i) - a(y_i)}{\max\{a(y_i), b(y_i)\}} \tag{9}$$

where $a(y_i)$ is the average dissimilarity of $y_i \in S_k$ from all other entities in its cluster S_k, and $b(y_i)$ the lowest average dissimilarity of the y_i from another cluster S_l at $l \neq k$. The larger the $S(y_i)$, the better the entity y_i sits in its cluster. The silhouette width of the partition is the sum of all $S(y_i)$ over all $i \in I$.

We apply the concept of silhouette width over five different measures of dissimilarity between vectors $x = (x_v)$ and $y = (y_v)$:

1. Squared Euclidean distance $d(x, y) = \sum_v (x_v - y_v)^2$;
2. Cosine $1 - c(x, y)$ where $c(x, y) = \sum_v x_v y_v / \sqrt{\sum_v x_v^2} \sqrt{\sum_v y_v^2}$;
3. Correlation $1 - r(x, y)$ where $r(x, y) = \sum_v (x_v - \bar{x})(y_v - \bar{y}) / \sqrt{\sum_v (x_v - \bar{x})^2} \sqrt{\sum_v (y_v - \bar{y})^2}$;
4. Power p of Minkowski p-distance $d_p^p(x, y) = \sum_v |x_v - y_v|^p$ where p is the same as in the tested run of iMWK-Means; and
5. A p-related analogue to the cosine dissimilarity defined as

$$c_p(x, y) = \sum_v \left| \frac{x_v}{\sqrt[p]{\sum_{v=1}^V |x_v|^p}} \frac{y_v}{\sqrt[p]{\sum_{v=1}^V |y_v|^p}} \right|^p \tag{10}$$

This definition is based on an analogue to the well-known equation relating the squared Euclidean distance and cosine, $d(x', y') = 2 - 2c(x', y')$, where $x' = \frac{x}{||x||}$, $y' = \frac{y}{||y||}$ are normed versions of the vectors. We select the p with the highest sum of silhouette widths.

4 Experiments

To validate the Minkowski exponent selection methods we experiment with, we use both real-world and synthetic datasets. The six real-world datasets taken from the UCI Irvine repository [2] are those that have been used by Huang et al. [3,9] as well as by Amorim and Mirkin [7]. Also, versions of these datasets obtained by adding uniformly random features are used to see the impact of the noise on the recovery of the Minkowski exponent.

The datasets are:

1. Iris. This dataset contains 150 flower specimens over four numerical features; it is partitioned in three groups. We devised two more Iris dataset versions by adding, respectively, extra two and extra four noise features. These are uniformly random.
2. Wine. This dataset contains 178 wine specimens partitioned in three groups and characterized by 13 numerical features that are chemical analysis results. We also use two more datasets by adding 7 and 13 noise features, respectively.
3. Hepatitis. This dataset contains 155 cases over 19 features, some of them categorical, partitioned in two groups. Two more versions of this dataset have been obtained by adding, in respect, 10 and 20 noise features.
4. Pima Indians Diabetes. This dataset contains 768 cases over 8 numerical features, partitioned in two groups. Two more versions of this dataset have been obtained by adding, in respect, 4 and 8 noise features.
5. Australian Credit Card Approval. This dataset contains 690 cases partitioned in two groups, originally with 15 features, some of them categorical. After a pre-processing step, described later in this section, we had a total of 42 numerical features.
6. Heart Disease. This dataset contains 270 cases partitioned in two groups referring to the presence or absence of a heart disease. This dataset has originally 14 features, including categorical ones. After the pre-processing step, there are 32 features in total.

Our synthetic data sets are of three formats: (F1) 1000x8-5: 1000 entities over 8 features consisting of 5 Gaussian clusters; (F2) 1000x15-7: 1000 entities over 15 features consisting of 7 Gaussian clusters; (F3) 1000x60-7: 1000 entities over 60 features consisting of 7 Gaussian clusters.

All the generated Gaussian clusters are spherical so that the covariance matrices are diagonal with the same diagonal value σ^2 generated at each cluster randomly between 0.5 and 1.5, and all centroid components independently generated from the Gaussian distribution with zero mean and unity variance. Cluster cardinalities are generated uniformly random, with a constraint that each generated cluster has to have at least 20 entities.

We standardize all datasets by subtracting the feature average from all its values, and dividing the result by half the feature's range. The standardization of categorical features follows a process described by Mirkin [12] to allow us to remain within the original K-Means framework. In this, each category is represented by a new binary feature, by assigning 1 to each entity which falls in the category and zero, otherwise. We then standardize these binary features by subtracting their grand mean, that is, the category's frequency. By adopting this method the centroids are represented by the proportions and conditional proportions rather than modal values.

Since all class pre-specified labels are known to us, we are able to map the clusters generated by iMWK-Means using a confusion matrix. We calculate the accuracy as the proportion of entities correctly clustered by each algorithm.

Table 1 Results of semi- and fully supervised experiments with the real-world datasets and their noisy versions

	Semi-supervised at different q						Supervised	
	5 %		15 %		25 %		100 %	
	Acc	p	Acc	p	Acc	p	Acc	p
Iris	93.09/6.07	1.14/0.53	95.15/1.94	1.23/0.62	95.60/1.75	1.16/0.44	96.67	1.1
Iris+2	93.55/2.19	1.150.52	94.95/1.97	1.13/0.28	95.81/1.53	1.10/0.06	96.67	1.1
Iris+4	93.39/3.89	1.20/0.73	94.76/1.46	1.10/0.09	95.19/1.21	1.13/0.11	96.00	1.1
Wine	87.64/6.30	1.48/0.84	91.88/2.28	1.88/1.06	92.45/1.09	2.13/1.19	93.82	1.6
Wine+7	89.22/4.51	1.13/0.22	91.81/1.77	1.44/0.50	92.85/1.46	1.55/0.48	94.38	2.2
Wine+13	90.18/7.40	1.21/0.53	93.24/1.73	1.13/0.10	93.63/1.44	1.13/0.09	94.38	1.1
Hepatitis	65.12/8.78	1.69/0.75	72.65/3.56	2.44/0.73	73.32/2.17	2.61/0.91	74.84	2.1
Hepatitis+10	69.16/9.13	2.31/1.52	76.59/7.74	3.46/1.24	80.10/4.09	3.90/1.04	82.58	4.3
Hepatitis+20	74.85/9.14	2.2/1.24	81.44/4.29	2.80/1.08	83.25/3.17	2.99/0.80	85.81	3.1
Pima	64.09/4.60	3.2/1.29	67.67/2.33	4.28/0.89	68.09/1.73	4.48/0.57	69.14	4.9
Pima+4	65.86/1.44	2.51/1.14	66.47/1.13	2.61/1.14	66.70/0.97	2.45/0.92	67.71	1.8
Pima+8	66.51/2.90	2.28/0.95	68.06/1.39	2.11/0.72	68.74/1.14	1.93/0.41	69.66	1.8
Austral CC	83.81/3.36	1.66/0.62	84.76/1.14	1.59/0.50	85.07/1.06	1.41/0.42	85.51	1.2
Heart	80.00/5.09	2.27/0.72	82.46/2.58	2.52/0.43	82.61/2.43	2.51/0.43	83.70	2.7

The accuracy and exponent shown are the averages and, after slash, standard deviations, over $R = 50$ runs

4.1 Results for the Semi-supervised Settings

Table 1 presents the results of our experiments for the semi-supervised setting at the real-world data. Different proportions of the labelled data are assumed to be known: $q = 5$, 10, 15, 20 and 25 %. For the purposes of comparison, the table also contains the results of a fully supervised experiment at which $q = 100$ %—the maximum accuracy. We have obtained these by running iMWK-Means for every p from the range of 1 to 5 in steps of 0.1 and checking which one had the highest accuracy using all the class labels at each dataset. The results using the semi-supervised selection of p show that at $q = 5$ % and $q = 10$ % an increase of 5 % in the size of the learning data amounts to an increase of about 1 % in the accuracy, with the accuracy reaching, at $q = 15$ %, to about 1–2 % within the maximum. Further increases of q to 20 and 25 % bring almost no increase in the accuracy, except for the case of the noisy Hepatitis data.

The results of our experiments using the semi-supervised algorithm on the synthetic datasets, shown in Table 2, present the same pattern as in Table 1. An increase of 5 % in the learning data produces an increase of around 1 % in the final accuracy of the algorithm till $q = 15$ %. In these, even with only 5 % of the data having been labelled, the algorithm still can reach accuracy of within 1–2 % of the maximum possible. Moreover, the exponent values stabilize from the very beginning at $q = 5$ %. This is an improvement over the real-world datasets, probably, because of a more regular structure of the synthetic datasets.

Table 2 Semi-supervised setting experiments with synthetic datasets

	Semi-supervised at different q						MCI	
	5 %		15 %		25 %		100 %	
	Acc	p	Acc	p	Acc	p	Acc	p
1000x8-5	80.45/5.34	3.79/0.63	82.40/3.91	3.81/0.64	82.75/3.67	3.82/0.63	82.94	3.83
1000x15-7	92.60/7.11	2.37/0.66	94.56/4.98	2.40/0.49	94.69/4.96	2.44/0.48	94.88	2.45
1000x60-7	99.19/3.09	1.56/0.41	99.87/1.43	1.48/0.22	100.0/0.00	1.47/0.19	100.0	1.48

The accuracy and exponent shown are the averages, accompanied by the standard deviations, over 50 runs for each of 10 Gaussian Model generated datasets

Table 3 The values of Minkowski exponent p and the accuracy obtained at the maximum of the silhouette width and minimum of MCI at the unsupervised experiments with the real-world datasets and their noisy versions; the maximum achievable accuracies are in the column on the left

			Silhouette width											
	Max		Sq. Euclid.		Cos		Corr.		Mink		Cos_p		MCI	
	Acc	p	Acc	p	Acc	p	Acc	p	Acc	p	Acc	p	Acc	p
Iris	96.67	1.1	93.33	3.7	90.67	5.0	94.00	3.4	96.00	1.3	90.67	2.3	96.67	1.1
Iris+2	96.67	1.1	84.00	4.4	87.33	3.7	90.00	1.8	96.67	1.1	90.00	1.8	96.67	1.1
Iris+4	96.00	1.1	72.00	4.7	72.00	4.7	72.00	4.7	96.00	1.1	93.33	1.9	95.33	1.4
Wine	93.82	1.6	93.82	1.6	93.82	1.6	92.13	2.2	92.70	1.4	92.13	2.2	90.45	1.2
Wine+7	94.38	2.2	93.26	2.0	91.57	1.9	90.45	2.5	92.70	1.3	92.13	2.3	89.89	1.2
Wine+13	94.38	1.1	93.82	1.3	93.26	1.6	89.89	2.2	94.38	1.1	92.13	1.9	93.82	1.3
Hepatitis	74.84	2.1	70.32	2.2	70.97	4.8	70.97	4.8	47.10	1.1	47.10	2.9	74.19	2.4
Hepatitis+10	82.58	4.3	81.94	5.0	63.23	3.6	63.23	3.6	76.13	1.4	62.58	1.8	52.9	1.9
Hepatitis+20	85.81	3.1	80.65	5.0	75.48	4.2	75.48	4.2	74.84	1.1	47.10	2.7	79.35	1.5
Pima	69.14	4.9	67.58	4.3	65.49	2.8	60.81	3.6	67.58	4.3	65.76	2.3	57.55	1.4
Pima+4	67.71	1.8	64.06	4.5	66.28	2.8	64.45	2.0	66.02	1.9	66.02	1.9	60.94	1.4
Pima+8	69.66	1.8	63.93	4.8	65.76	1.9	65.76	1.9	65.76	1.9	65.10	4.5	68.49	1.5
Aust CC	85.51	1.2	85.51	1.2	85.55	1.2	85.55	1.2	85.51	1.2	73.33	3.8	78.84	2.4
Heart	83.70	2.7	83.33	2.6	83.33	2.6	83.33	2.6	75.19	1.1	83.33	2.6	75.19	1.9

4.2 Results at the Unsupervised Setting

In this set of experiments there is no learning stage. Because of this we found it reasonable to take p in the interval from 1.1 to 5 rather than 1 to 5, still in steps of 0.1. At $p = 1$ iMWK-Means selects a single feature from the dataset [7] putting a weight of zero in all others. In our view, there are only a few scenarios in which the optimal p would be 1 and these would be very hard to find without learning data.

Table 3 presents the results of our experiments with the silhouette width for five dissimilarity measures; three Euclidean: squared distance, cosine, correlation, and two Minkowski's: distance and cosine. When using the latter two, the exponent p is the same as the one used in the iMWK-Means clustering.

For the sake of comparison, the Table 3 also presents the maximum accuracy for each dataset. The table shows that it is indeed possible to select a good p

Table 4 The values of Minkowski exponent p and the accuracy obtained at different rules defined above

	Max		Sq. Euclid.		Mink		MCI		Consensus	
	Acc	p	Acc	p	Acc	p	Acc	p	p	Acc
Iris	96.67	1.1	93.33	3.7	96.00	1.3	96.67	1.1	1.20	96.00
Iris+2	96.67	1.1	84.00	4.4	96.67	1.1	96.67	1.1	1.10	96.67
Iris+4	96.00	1.1	72.00	4.7	96.00	1.1	95.33	1.4	1.25	94.67
Wine	93.82	1.6	93.82	1.6	92.70	1.4	90.45	1.2	1.30	92.13
Wine+7	94.38	2.2	93.26	2.0	92.70	1.3	89.89	1.2	1.25	89.33
Wine+13	94.38	1.1	93.82	1.3	94.38	1.1	93.82	1.3	1.30	93.83
Hepatitis	74.84	2.1	70.32	2.2	47.10	1.1	74.19	2.4	2.30	60.00
Hepatitis+10	82.58	4.3	81.94	5.0	76.13	1.4	52.90	1.9	1.65	70.97
Hepatitis+20	85.81	3.1	80.65	5.0	74.84	1.1	79.35	1.5	5.00	80.65
Pima	69.14	4.9	67.58	4.3	67.58	4.3	57.55	1.4	4.30	67.58
Pima+4	67.71	1.8	64.06	4.5	66.02	1.9	60.94	1.4	1.65	66.41
Pima+8	69.66	1.8	63.93	4.8	65.76	1.9	68.49	1.5	1.70	69.53
Aust CC	85.51	1.2	85.51	1.2	85.51	1.2	78.84	2.4	1.20	85.51
Heart	83.70	2.7	83.33	2.6	75.19	1.1	75.19	1.9	1.50	75.19

without having labels for a given dataset. For example, silhouette width based on Minkowski distance works well on Iris and Heart, based on Euclidean squared distance, on Wine, Pima, and Heart, and MCI works well on Hepatitis. Overall, the best performance has shown the MCI as it has the best worst case scenario among all the indexes under consideration. Its highest difference between its accuracy and the maximum possible is equal to $-12.24\,\%$ (on Pima), whereas the other indexes lead to the highest difference between 24 and $38.71\,\%$. Yet none of the indexes is reliable enough to be used with no reservations. Taking into account the fact that different indexes lead to different solutions, one may suggest using a consensus rule (Table 4).

Specifically, let us take the MCI index and two silhouette width indexes, that are based on the Euclidean squared distance (SWE) and that are based on Minkowski distance (SWM), and, when they are in disagreement, use the value of p that is the average of those two that are in agreement (see Table 4).When iMWK-Means is unable to find the required number of clusters using the p agreed between two indexes, as with the Hepatitis + 20 at $p = 1.3$, we use the p from the remaining index in our experiments

The results of experiments in the unsupervised setting at the synthetic data sets are presented in Table 5. At the synthetic datasets we can observe a different pattern. The MCI is no longer the most promising algorithm to select p. In these the cosine and correlation are those recovering the adequate ps and, thus, having considerably better results.

Table 5 Results of searching for the best p at the unsupervised setting

			Silhouette width											
	Max		Sq. Euclid.		Cos		Corr.		Mink		Cos_p		MCI	
	Acc	p	Acc	p	Acc	p	Acc	p	Acc	p	Acc	p	Acc	p
1000x8-5	82.94	3.83	81.79/4.20	3.80/0.73	81.81/4.18	3.81/0.73	81.65/4.16	3.58/0.60	81.80/4.19	3.82/0.73	69.59/13.44	2.47/0.67	71.70/10.41	5.0/0.00
1000x15-7	94.88	2.45	92.28/8.77	2.63/0.60	93.47/7.42	2.46/0.37	93.47/7.42	2.46/0.37	92.00/8.91	2.91/0.86	93.22/8.38	2.16/0.15	76.49/7.98	4.99/0.03
1000x60-7	100.0	1.48	99.83/0.54	1.80/1.10	100.0/0.00	1.48/0.20	100.0/0.00	1.48/0.20	99.84/0.51	2.08/0.37	98.58/2.93	2.65/0.72	90.50/8.30	5.0/0.00

The accuracy and exponent values are the averages and standard deviations over each of the 10 GMs

5 Conclusion

The use of weights in K-Means clustering has shown good results [3, 8–11], in particular when utilizing the Minkowski distance metric [6, 7]. Its version oriented at determining the number of clusters and the initial centroids, the intelligent Minkowski Weighted K-Means showed considerably better accuracy results, at an appropriate Minkowski exponent p, than a number of other algorithms [7]. However, finding an appropriate p remained an open issue. This paper presents a study regarding the amount of labelled data necessary for a good recovery of p under a semi-supervised approach, as well as an unsupervised method based on indexes of correspondence between the found partition and the dataset structure.

We have found that in most datasets it is possible to recover a good p with as low as 5 % of the data being labelled, and that reasonable results can be obtained by using individual indexes over the clustering, the MCI or silhouette width indexes, or a combined "consensus" rule. It is quite likely that these findings can be relevant for the Minkowski partition around medoids algorithm [5].

However, the iMWK-Means algorithm may have difficulties finding appropriate weights for datasets containing informative but redundant features. In this case the algorithm sets the weights of all such features to high values instead of removing some features by setting some weights to zero. This is an issue that we intend to address in future research.

References

1. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recognit. **46**, 243–256 (2012)
2. Bache, K., Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2013)
3. Chan, E.Y., Ching, W.K., Ng, M.K., Huang, J.Z.: An optimization algorithm for clustering using weighted dissimilarity measures. Pattern Recognit. **37**(5), 943–952 (2004)
4. Chiang, M.M.T., Mirkin, B.: Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. J. Classif. **27**(1), 3–40 (2010)
5. de Amorim, R.C., Fenner, T.: Weighting features for partition around medoids using the minkowski metric. In: Jaakko, H., Frank, K., Allan, T. (eds.) Advances in Intelligent Data Analysis. Lecture Notes in Computer Science, vol. 7619, pp. 35–44. Springer, Berlin (2012)
6. de Amorim, R.C., Komisarczuk, P.: On initializations for the minkowski weighted k-means. In: Jaakko, H., Frank, K., Allan, T. (eds.) Advances in Intelligent Data Analysis. Lecture Notes in Computer Science, vol. 7619, pp. 45–55. Springer, Berlin (2012)
7. de Amorim, R.C., Mirkin, B.: Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. Pattern Recognit. **45**(3), 1061–1075 (2012)
8. Frigui, H., Nasraoui, O.: Unsupervised learning of prototypes and attribute weights. Pattern Recognit. **37**(3), 567–581 (2004)
9. Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 657–668 (2005)
10. Huang, J.Z., Xu, J., Ng, M., Ye, Y.: Weighting method for feature selection in k-means. In: Computational Methods of Feature Selection, pp. 193–209. Chapman & Hall, London (2008)

11. Makarenkov, V., Legendre, P.: Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. J. Classif. **18**(2), 245–271 (2001)
12. Mirkin, B.: Clustering for Data Mining: A Data Recovery Approach, vol. 3. Chapman & Hall, London (2005)
13. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)

High-Dimensional Data Classification

Vijay Pappu and Panos M. Pardalos

We dedicate this paper to the 70th birthday of our colleague and friend Dr. Boris Mirkin

Abstract Recently, high-dimensional classification problems have been ubiquitous due to significant advances in technology. High dimensionality poses significant statistical challenges and renders many traditional classification algorithms impractical to use. In this chapter, we present a comprehensive overview of different classifiers that have been highly successful in handling high-dimensional data classification problems. We start with popular methods such as Support Vector Machines and variants of discriminant functions and discuss in detail their applications and modifications to several problems in high-dimensional settings. We also examine regularization techniques and their integration to several existing algorithms. We then discuss more recent methods, namely the hybrid classifiers and the ensemble classifiers. Feature selection techniques, as a part of hybrid classifiers, are introduced and their relative merits and drawbacks are examined. Lastly, we describe AdaBoost and Random Forests in the ensemble classifiers and discuss their recent surge as useful algorithms for solving high-dimensional data problems.

Keywords High dimensional data classification • Ensemble methods • Feature selection • Curse of dimensionality • Regularization

V. Pappu (✉) • P.M. Pardalos
Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA
e-mail: vijay.s.pappu@aexp.com

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_8,

1 Introduction

In the past decade, technological advances have had a profound impact on society and the research community [45]. Massive amounts of high-throughput data can be collected simultaneously and at relatively low cost. Often, each observation is characterized with thousands of variables/features. For example, in biomedical studies, huge numbers of magnetic resonance images (MRI) and functional MRI data are collected for each subject [66]. The data collected from gene expression microarrays consist of thousands of genes that constitute features [17]. Various kinds of spectral measurements including Mass Spectroscopy and Raman Spectroscopy are very common in chemometrics, where the spectra are recorded in channels that number well into the thousands [30, 80]. Satellite imagery has been used in natural resource discovery and agriculture, collecting thousands of high-resolution images. Examples of these kinds are plentiful in computational biology, climatology, geology, neurology, health science, economics, and finance among others. In several applications, the measurements tend to be very expensive and hence the number of samples in many datasets are on the order of tens, or maybe low hundreds. These datasets, often called the high-dimension low-sample size (HDLSS) datasets, are characterized with a large number of features p and a relatively small number of samples n; with $p >> n$ [98]. These massive collections of data along with many new scientific problems create golden opportunities and significant challenges for the development of mathematical sciences.

Classification is a supervised machine learning technique that maps some combination of input variables, which are measured or preset, into predefined classes. Classification problems occur in several fields of science and technology like discriminating cancerous cells from non-cancerous cells, web document classification, categorizing images in remote sensing applications among many others. Several algorithms starting from Neural Networks [44], Logistic Regression [57], linear discriminant analysis (LDA) [64], support vector machines (SVM) [92] and more recently ensemble methods like Boosting [33] and Random Forests [8], have been proposed to solve the classification problem in different contexts. However, the availability of massive data along with new scientific problems arising in the fields of computational biology, microarray gene expression analysis, etc., have reshaped statistical thinking and data analysis. The high-dimensional data has posed significant challenges to standard statistical methods and have rendered many existing classification techniques impractical [53]. Hence, researchers have proposed several novel techniques to handle the inherent difficulties of high-dimensional spaces that are discussed below.

1.1 Statistical Challenges of High-Dimensional Data Spaces

1.1.1 Curse of Dimensionality

The accuracy of classification algorithms tends to deteriorate in high dimensions due to a phenomenon called the *curse of dimensionality* [27, 60]. This phenomenon is illustrated by Trunk [90] using an example in [90]. Trunk found that (1) the best test error was achieved using a finite number of features; (2) using an infinite number of features, test error degrades to the accuracy of random guessing; and (3) the optimal dimensionality increases with increasing sample size. Also, a naive learning technique (dividing the attribute space into cells and associating a class label with each cell) that predicts using a majority voting scheme requires the number of training samples to be an exponential function of the feature dimension [50]. Thus, the ability of an algorithm to converge to a true model deteriorates rapidly as the feature dimensionality increases.

1.1.2 Poor Generalization Ability

A further challenge for modeling in high-dimensional spaces is to avoid overfitting the training data [17]. It is important to build a classification model with good generalization ability. It is expected that such a model, in addition to performing well on the training set, would also perform equally well on an independent testing set. However, often the small number of samples in high-dimensional data settings cause the classification model to overfit to the training data, thereby having poor generalization ability for the model. Two of the more common approaches to addressing these challenges of high-dimensional spaces are reducing the dimensionality of the dataset or applying methods that are independent of data dimensionality. We discuss several classifiers pertaining to these two approaches in subsequent sections.

In this survey, we present several state-of-the-art classifiers that have been very successful for classification tasks in high-dimensional data settings. The remainder of the chapter is organized as follows. Section 2 talks about SVM and its variants. Discriminant functions and their modifications including regularized techniques are discussed in Sect. 3. Section 4 discusses hybrid classifiers that include several feature selection techniques combined with other traditional classification algorithms. Recent developments in ensemble methods and their applications to high-dimensional data problems are discussed in Sect. 5. Some software packages implementing the methods in different programming languages are discussed in Sect. 6. Concluding remarks are presented in Sect. 7.

2 Support Vector Machines

2.1 Hard-Margin Support Vector Machines

In the last decade, SVM [92] have attracted the attention of many researchers with successful application to several classification problems in bioinformatics, finance and remote sensing among many others [13, 69, 89]. Standard SVM construct a hyperplane, also known as *decision boundary*, that *best* divides the input space χ into two disjoint regions. The hyperplane $f : \chi \rightarrow \Re$, is estimated from the training set S. The class membership for an unknown sample $\boldsymbol{x} \in \chi$ can be based on the classification function $g(\boldsymbol{x})$ defined as:

$$g(\boldsymbol{x}) = \begin{cases} -1, & f(\boldsymbol{x}) < 0 \\ 1, & f(\boldsymbol{x}) > 0 \end{cases} \tag{1}$$

Consider a binary classification problem with the training set S defined as:

$$S = \{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \Re^p, y_i \in \{-1, 1\}\}, \quad i = 1, 2, \ldots, n \tag{2}$$

where y_i is either -1 or 1 depending on the class that each $\boldsymbol{x}_i$ belongs to. Assume that the two classes are linearly separable and hence there exists atleast one hyperplane that separates the training data correctly. A hyperplane parameterized by the normal vector $\boldsymbol{w} \in \Re^p$ and bias $b \in \Re$ is defined as:

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle - b = 0 \tag{3}$$

where the inner product $\langle \cdot, \cdot \rangle$ is defined on $\Re^p \times \Re^p \rightarrow \Re$. The training set S satisfies the following linear inequality with respect to the hyperplane:

$$y_i (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b) \geq 1 \quad \forall i = 1, 2, \ldots, n \tag{4}$$

where the parameters $\boldsymbol{w}$ and b are chosen such that the distance between the hyperplane and the closest point is maximized. This geometrical margin can be expressed by the quantity $\frac{1}{||\boldsymbol{w}||}$. Hence, for linearly separable set of training points, SVM can be formulated a linearly constrained quadratic convex optimization problem given as:

$$\begin{aligned} &\underset{\boldsymbol{w}, b}{\text{minimize}} \quad ||\boldsymbol{w}||_2^2 \\ &\text{subject to} \quad y_i (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b) \geq 1 \quad \forall i = 1, 2, \ldots, n \end{aligned} \tag{5}$$

This classical convex optimization problem can be rewritten (using the Lagrangian formulation [5]) into the following dual problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\Re^n}{\text{maximize}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle) \\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0, \quad \text{and}, \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, n \end{aligned} \tag{6}$$

where the Lagrange multipliers α_i $(i = 1, 2, \ldots, n)$ expressed in (6) can be estimated using quadratic programming (QP) methods [22]. The optimal hyperplane f can then be estimated using the Lagrange multipliers obtained from solving (6) and the training samples, i.e.,

$$f(\boldsymbol{x}) = \sum_{i\in S'}\alpha_i y_i(\langle \boldsymbol{x}\cdot\boldsymbol{x}_i\rangle) - b \tag{7}$$

where S' is the subset of training samples called *support vectors* that correspond to non-zero Lagrange multipliers α_i. Support vectors include the training points that exactly satisfy the inequality in (5) and lie at a distance equal to $\frac{1}{\|\boldsymbol{w}\|}$ from the optimal separating hyperplane. Since the Lagrange multipliers are non-zero only for the *support vectors* and zero for other training samples, the optimal hyperplane in (7) effectively consists of contributions from the *support vectors*. It is also important to note that the Lagrange multipliers α_i qualitatively provide relative weight of each *support vector* in determining the optimal hyperplane.

The convex optimization problem in (5) and the corresponding dual in (6) converge to a global solution *only* if the training set is linearly separable. These SVM are called *hard-margin support vector machines*.

2.2 Soft-Margin Support Vector Machines

The *maximum-margin* objective introduced in the previous subsection to obtain the *optimal* hyperplane is susceptible to the presence of outliers. Also, it is often difficult to adhere to the assumption of linear separability in real-world datasets. Hence, in order to handle nonlinearly separable datasets as well as be less sensitive to outliers, *soft-margin support vector machines* are proposed. The objective cost function in (5) is modified to represent two competing measures namely, *margin maximization* (as in the case of linearly separable data) and *error minimization* (to penalize the wrongly classified samples). The new cost function is defined as:

$$\Psi(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i=1}^{n}\xi_i \tag{8}$$

where $\boldsymbol{\xi}$ is the *slack variable* introduced to account for the non-separability of data, and the constant C represents a regularization parameter that controls the penalty assigned to errors. The larger the C value, the higher the penalty associated to misclassified samples. The minimization of the cost function expressed in (8) is subject to the following constraints:

$$\begin{aligned} y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b) &\geq 1 - \xi_i, \quad \forall i = 1, 2, \ldots, n \\ \xi_i &\geq 0, \quad \forall i = 1, 2, \ldots, n \end{aligned} \tag{9}$$

The convex optimization problem can then be formulated using (8) and (9) for the nonlinearly separable data as:

$$\begin{aligned} &\underset{\boldsymbol{w}, b, \boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}||\boldsymbol{w}||_2^2 + C \sum_{i=1}^{n} \xi_i \\ &\text{subject to} \quad y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, 2, \ldots, n \end{aligned} \tag{10}$$

The optimization problem in (10) accounts for the outliers by adding a penalty term $C\xi_i$ for each outlier to the objective function. The corresponding dual to (10) can be written using the Lagrange formulation as:

$$\begin{aligned} &\underset{\boldsymbol{\alpha} \in \Re^n}{\text{maximize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle) \\ &\text{subject to} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \text{and,} \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n \end{aligned} \tag{11}$$

The quadratic optimization problem in (11) can be solved using standard QP techniques [22] to obtain the Lagrange multipliers α_i.

2.3 Kernel Support Vector Machines

The idea of linear separation between two classes mentioned in the subsections above can be naturally extended to handle nonlinear separation as well. This is achieved by mapping the data through a particular nonlinear transformation into a higher dimensional feature space. Assuming that the data is linearly separable in this high dimensional space, a linear separation, similar to earlier subsections, can be found. Such a hyperplane can be achieved by solving a similar dual problem defined in (11) by replacing the inner products in the original space with inner products in the transformed space. However, an explicit transformation from the original space to feature space could be expensive and at times infeasible as well. The kernel method [12] provides an elegant way of dealing with such transformations.

Consider a kernel function $K(\cdot,\cdot)$, satisfying *Mercer's* theorem, that equals an inner product in the transformed higher dimensional feature space [65], i.e.,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle \tag{12}$$

where $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$ correspond to the mapping of data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the original space to the feature space. There are several kernel functions defined in literature that satisfy *Mercer's* conditions. One such kernel, called the Gaussian kernel is given by:

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\sigma ||\boldsymbol{x}_i - \boldsymbol{x}||^2) \tag{13}$$

where σ is a parameter inversely proportional to the width of the Gaussian radial basis function. Another extensively studied kernel is the polynomial function of order p expressed as

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = (\langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + 1)^p \tag{14}$$

Such kernel functions defined above allow for efficient estimation of inner products in feature spaces without the explicit functional form of the mapping Φ. This elegant calculation of inner products in higher dimensional feature spaces, also called the *kernel trick*, considerably simplifies the solution to the dual problem. The inner products between the training samples in the dual formulation (11) can be replaced with a kernel function K and rewritten as:

$$\begin{aligned} \underset{\boldsymbol{\alpha} \in \Re^n}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \text{and,} \quad 0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, n \end{aligned} \tag{15}$$

The *optimal* hyperplane f obtained in the higher dimensional feature space can be conveniently expressed as a function of data in the original input space as:

$$f(\boldsymbol{x}) = \sum_{i \in S'} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) - b \tag{16}$$

where S' is a subset of training samples with non-zero Lagrange multipliers α_i. The shape of $f(\boldsymbol{x})$ depends on the type of kernel functions adopted.

It is important to note that the performance of kernel-based SVM is dependent on the optimal selection of multiple parameters, including the kernel parameters (e.g., σ and p parameters for the Gaussian and polynomial kernels, respectively) and the regularization parameter C. A simple and successful technique that has been employed involves a grid search over a wide range of the parameters.

The classification accuracy of SVM for every pair of parameters is estimated using a leave-one-out cross-validation technique and the pair corresponding to the highest accuracy is chosen. Also, some interesting automatic techniques have been developed to estimate these parameters [15, 16]. They involve constructing an optimization problem that would maximize the margin as well as minimize the estimate of the expected generalization error. Optimization of the parameters is then carried out using a gradient descent search over the space of the parameters. Recently, more heuristic-based approaches have been proposed to deal with this issue. A continuous version of Simulated Annealing (SA) called *Hide and Seek SA* was employed in [61] to estimate multiple parameters as well as select a subset of features to improve the classification accuracy. Similar approaches combining particle swarm optimization (PSO) with SVM are proposed in [39,62]. Furthermore, a modified Genetic Algorithm (GA) was also implemented along with SVM to estimate the optimal parameters [47].

2.4 SVM Applied to High-Dimensional Classification Problems

Support vector machines have been successfully applied to high-dimensional classification problems arising in fields like remote sensing, web document classification, microarray analysis etc. As mentioned earlier, conventional classifiers like logistic regression, maximum likelihood classification etc., on high-dimensional data tend to overfit the model using training data and run the risk of achieving lower accuracies on testing data. Hence, a pre-processing step like either feature selection and/or dimensionality reduction techniques are proposed to alleviate the problem of *curse of dimensionality* while working with these traditional classifiers. Surprisingly, SVM have been successfully applied to hyperspectral remote sensing images without any pre-processing steps [69]. Researchers show that SVM are more effective that the traditional pattern recognition approach that involves a feature selection procedure followed by a conventional classifier and are also insensitive to Hughes phenomena [49]. This is particularly helpful as it avoids the unnecessary additional computation of an intermediary step like feature selection/dimensionality reduction to achieve high classification accuracy.

Similar observations were reported in the field of document classification in [52], where SVM were trained directly on the original high-dimensional input space. Kernel SVM (Gaussian and polynomial kernels) were employed and compared with other conventional classifiers like k-NN classifiers, Naive-Bayes Classifier, Rocchio Classifier and C4.5 Decision Tree Classifier. The results show that Kernel SVM outperform the traditional classifiers. Also, in the field of microarray gene expression analysis, SVM have been successfully applied to perform classification of several cancer diagnosis tasks [9, 74].

The insensitivity of SVM to overfitting and the ability to overcome the curse of dimensionality can be explained via the generalization error bounds developed by Vapnik et al. [93]. Vapnik showed the following generalization error bounds for Large Margin Classifiers:

$$\epsilon = \tilde{O}\left(\frac{1}{m}\left(\frac{R^2}{\gamma^2} + \log\frac{1}{\delta}\right)\right) \tag{17}$$

where m is the number of training samples, γ is the margin between the parallel planes, and $(R, \delta) \in \Re^+$ with $0 < \delta \leq 1$. This error bound is inversely dependent on the sample size m and the margin γ. For a finite sample size, maximizing the margin γ (or minimizing the weight vector) would reduce the generalization error ϵ. Interestingly, this error bound does *not* depend on the dimensionality of the input space. Since, it is highly likely to linearly separate the data in higher dimensions, SVM tend to perform well with classification tasks in high dimensions.

3 Discriminant Functions

A discriminant function $g : \Re^p \rightarrow \{-1, 1\}$ assigns either class 1 or class 2 to an input vector $\boldsymbol{x} \in \Re^p$. We consider here a class of discriminant functions $\mathcal{G}$ that are well studied in literature and traditionally applied to binary classification problems.

3.1 Quadratic and Linear Discriminant Analysis

Consider a binary classification problem with classes $\mathcal{C}_1$ and $\mathcal{C}_2$ and prior probabilities given as π_1 and π_2. Assume the class conditional probability densities $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ to be normally distributed with mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, respectively:

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\right) \quad k = 1, 2. \tag{18}$$

where, $|\boldsymbol{\Sigma}_k|$ is the determinant of the covariance matrix $\boldsymbol{\Sigma}_k$. Following *Bayes* optimal rule [3], *quadratic discriminant analysis (QDA)* [64] assigns class 1 to an input vector $\boldsymbol{x}$ if the following condition holds:

$$\pi_1 f_1(\boldsymbol{x}) \geq \pi_2 f_2(\boldsymbol{x}) \tag{19}$$

Linear discriminant analysis [64] further assumes the covariances $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are equal to $\boldsymbol{\Sigma}$ and classifies an input vector again in accordance to *Bayes* optimal rule. The condition in (19) can then be rewritten as:

$$\log\frac{\pi_1}{\pi_2} + (\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \geq 0, \quad \boldsymbol{\mu} = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2). \tag{20}$$

Assuming the prior probabilities to be equal, (20) is equivalent to:

$$(\boldsymbol{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_1) \leq (\boldsymbol{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_2) \tag{21}$$

It is interesting to note that LDA compares the *squared Mahalanobis distance* [21] of $\boldsymbol{x}$ from the class means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and assigns the class that is closest. The squared Mahalanobis distance of a point $\boldsymbol{x}$ from a distribution $\mathcal{P}$ characterized by mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is defined as:

$$d_M(\boldsymbol{x}, \mathcal{P}) = (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \tag{22}$$

This distance measure, unlike *Euclidean distance measure*, accounts for correlations among different dimensions of $\boldsymbol{x}$. Equation (21) shows how LDA differs from other *distance-based* classifiers like *k-NN* classifier [3] which measures Euclidean distance to assign the class.

3.2 Fisher Linear Discriminant Analysis

Fisher linear discriminant analysis (FLDA) [3], unlike LDA, does not make assumptions on the class conditional densities. Instead, it estimates the class means from the training set. In practice, the most commonly used estimators are their maximum-likelihood estimates, given by:

$$\hat{\boldsymbol{\mu}}_1 = \frac{1}{N_1} \sum_{k \in \mathcal{C}_1} \boldsymbol{x}_k, \qquad \hat{\boldsymbol{\mu}}_2 = \frac{1}{N_2} \sum_{k \in \mathcal{C}_2} \boldsymbol{x}_k. \tag{23}$$

Fisher linear discriminant analysis attempts to find a projection vector $\boldsymbol{w}$ that maximizes the class separation. In particular, it maximizes the following *Fisher criterion* given as:

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}} \tag{24}$$

where $\boldsymbol{S}_B$ is the *between-class* covariance matrix and is given by:

$$\boldsymbol{S}_B = (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \tag{25}$$

and $\boldsymbol{S}_W$ is the *within-class* covariance matrix and is given by:

$$\boldsymbol{S}_W = \sum_{k \in \mathcal{C}_1} (\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_1)^T + \sum_{k \in \mathcal{C}_2} (\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_2)(\boldsymbol{x}_k - \hat{\boldsymbol{\mu}}_2)^T \tag{26}$$

The optimal Fisher discriminant $\boldsymbol{w}^*$ can be obtained by maximizing the *Fisher criterion*:

$$\underset{\boldsymbol{w}}{\text{maximize}} \quad J(\boldsymbol{w}) \tag{27}$$

An important property to notice about the objective function $J(\boldsymbol{w})$ is that it is invariant to the rescalings of the vector $\boldsymbol{w} \rightarrow \alpha \boldsymbol{w}, \forall \alpha \in \Re$. Hence, $\boldsymbol{w}$ can be chosen in a way that the denominator is simply $\boldsymbol{w}^T S_W \boldsymbol{w} = 1$, since it is a scalar itself. For this reason, we can transform the problem of maximizing *Fisher criterion J* into the following constrained optimization problem,

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{maximize}} \quad & \boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w} \\ \text{subject to} \quad & \boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w} = 1 \end{aligned} \tag{28}$$

The KKT conditions for (28) can be solved to obtain the following generalized eigenvalue problem, given as:

$$\boldsymbol{S}_B \boldsymbol{w} = \lambda \boldsymbol{S}_W \boldsymbol{w} \tag{29}$$

where λ represents the eigenvalue and the optimal vector $\boldsymbol{w}^*$ corresponds to the eigenvector with the largest eigenvalue $\lambda_{\max}$ and is proportional to:

$$\boldsymbol{w}^* \propto \boldsymbol{S}_W^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \tag{30}$$

The class of an input vector $\boldsymbol{x}$ is determined using the following condition:

$$\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle < c \tag{31}$$

where $c \in \Re$ is a threshold constant.

3.3 Diagonal Linear Discriminant Analysis

Diagonal linear discriminant analysis (DLDA) extends on LDA and assumes independence among the features [35]. In particular, the discriminant rule in (20) is replaced with:

$$\log \frac{\pi_1}{\pi_2} + (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{D}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \geq 0 \tag{32}$$

where $\boldsymbol{D} = \text{diag}(\boldsymbol{\Sigma})$. The off-diagonal elements of the covariance matrix $\boldsymbol{\Sigma}$ are replaced with zeros by independence assumption.

Similarly, *diagonal quadratic discriminant analysis* (DQDA) [28] assumes the *independence rule* for QDA. The discriminant rule in this case is given by:

$$\log \frac{\pi_1}{\pi_2} + (\boldsymbol{x} - \boldsymbol{\mu}_2)^T \boldsymbol{D}_2^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_2) - (\boldsymbol{x} - \boldsymbol{\mu}_1)^T \boldsymbol{D}_1^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_1) \geq 0 \tag{33}$$

where $\boldsymbol{D}_1 = \text{diag}(\boldsymbol{\Sigma}_1)$, and $\boldsymbol{D}_2 = \text{diag}(\boldsymbol{\Sigma}_2)$.

Diagonal quadratic discriminant analysis and DLDA classifiers are sometimes called "naive Bayes" classifiers because they can arise in a Bayesian setting [2]. Additionally, it is important to note that FLDA and Diagonal Discriminant analysis (DLDA and DQDA) are commonly generalized to handle multi-class problems as well.

3.4 Sparse Discriminant Analysis

The optimal discriminant vector in FLDA (30) involves estimating the inverse of covariance matrix obtained from sample data. However the high dimensionality in some classification problems poses the threat of singularity and thus leads to poor classification performance. One approach to overcome singularity involves a variable selection procedure that selects a subset of variables most appropriate for classification. Such a *sparse* solution has several advantages including better classification accuracy as well as interpretability of the model. One of the ways to induce sparsity is via the path of regularization. Regularization techniques have been traditionally used to prevent overfitting in classification models, but recently, they have been extended to induce sparsity as well in high-dimensional classification problems. Here, we briefly discuss some standard regularization techniques that facilitate variable selection and prevent overfitting.

Given a set of instance-label pairs $(\boldsymbol{x}_i, y_i)$; $i = 1, 2, \ldots, n$; a regularized classifier optimizes the following unconstrained optimization problem:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \Phi(\boldsymbol{x}, y, \boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_p \tag{34}$$

where Φ represents a non-negative loss function, $(p, \lambda) \in \Re$ and $\boldsymbol{\beta}$ is the coefficient vector. Classifiers with $p = 1$ (*Lasso-penalty*) and $p = 2$ (*ridge-penalty*) have been successfully applied to several classification problems [99].

In a regression setting, Tibshirani [85] introduced variable selection via the framework of regularized classifiers using the l_1-norm. This method, also called *least absolute shrinkage and selection operator (LASSO)*, considers the least-squares error as the loss function. The user-defined parameter λ balances the regularization and the loss terms. The l_1-norm in Lasso produces some coefficients that are exactly 0 thus facilitating the selection of only a subset of variables useful for regression. The Lasso regression, in addition to providing a sparse model, also shares the stability of ridge regression. Several algorithms have been successfully

employed to solve the Lasso regression in the past decade. Efron et al. [29] showed that, starting from zero, the Lasso solution paths grow piecewise linearly in a predictable way and hence exploit this predictability to propose a new algorithm called *Least Angle Regression* that solves the entire Lasso path efficiently. The Lasso framework has been further extended to several classification problems by considering different loss functions, and has been highly successful in producing sparse models with high classification accuracy.

A Lasso-type framework, however, is not without its limitations. Zou and Hastie [99] mention that a Lasso framework, in high-dimensional problems, suffers from two drawbacks namely, the number of variables selected is limited by the number of samples n, and in the case of highly correlated features, the method selects one of them, neglecting the rest and also does not care about the one selected. The second limitation, also called the *grouping effect*, is very common in high-dimensional classification problems like microarray gene analysis where a group of variables are highly correlated to each other. The authors propose a new technique that overcomes the limitations of Lasso. The technique, called *elastic-net*, considers a convex combination of l_1 and l_2-norms to induce sparsity. In particular, in an *elastic-net* framework, the following optimization problem is minimized:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \Phi(\boldsymbol{x}, y, \boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1 + (1-\lambda)||\boldsymbol{\beta}||_2 \tag{35}$$

where Φ is the loss function, and $0 \leq \lambda \leq 1$. When $\lambda = 0$ (or $= 1$), the elastic-net framework simplifies to Lasso (or ridge) frameworks. The method could simultaneously perform variable selection along with continuous shrinkage and also select groups of correlated variables. An efficient algorithm, called LARS-EN, along the lines of LARS, was proposed to solve the elastic-net problem. It is important to note that these regularized frameworks are very general and can be added to models that suffer from overfitting. They provide better generalization performance by inherently performing variable selection and thus also producing better interpretable models.

Sparsity can be induced to the solution of FLDA using regularization techniques described above. One such method called *sparse linear discriminant analysis (SLDA)*, is inspired from *penalized least squares* where regularization is applied to the solution of least squares problem via *Lasso-penalty*. The penalized least squares problem is formulated as:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{36}$$

where $\boldsymbol{X}$ represents the data matrix and $\boldsymbol{y}$ is the outcome vector. The second term in (36) is assumed to induce sparsity to the optimal $\boldsymbol{\beta}$.

In order to induce sparsity in FLDA via the l_1 penalty, the generalized eigenvalue problem in (29) is first reformulated as an equivalent least squares regression problem and is shown that the optimal discriminant vector of FLDA is equivalent to

the optimal regression coefficient vector. This is achieved by applying the following theorem:

Theorem. *Assume the between-class covariance matrix* $\boldsymbol{S}_B \in \Re^{p\times p}$ *and the within-class covariance matrix* $\boldsymbol{S}_W \in \Re^{p\times p}$ *be given by* (25) *and* (26)*. Also, assume* $\boldsymbol{S}_W$ *is positive definite and denote its Cholesky decomposition as* $\boldsymbol{S}_W = \boldsymbol{R}_W^T \boldsymbol{R}_W$ *where* $\boldsymbol{R}_W \in \Re^{p\times p}$ *is an upper triangular matrix. Let* $\boldsymbol{H}_B \in \Re^{n\times p}$ *satisfy* $\boldsymbol{S}_B = \boldsymbol{H}_B^T \boldsymbol{H}_B$*. Let* $\boldsymbol{v}_1, \boldsymbol{v}_2 \ldots, \boldsymbol{v}_q (q \leq min(p, n-1))$ *denote the eigenvectors of problem* (29) *corresponding to the* q *largest eigenvalues* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q$*. Let* $\boldsymbol{A} \in \Re^{p\times q} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_q]$ *and* $\boldsymbol{B} \in \Re^{p\times q} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_q]$*. For* $\lambda > 0$*, let* $\hat{\boldsymbol{A}}$ *and* $\hat{\boldsymbol{B}}$ *be the solution to the following least squares regression problem:*

$$\begin{aligned} &\underset{\boldsymbol{A},\boldsymbol{B}}{\text{minimize}} \ \sum_{i=1}^{n} \|\boldsymbol{R}_W^{-T} \boldsymbol{H}_{B,i} - \boldsymbol{A}\boldsymbol{B}^T \boldsymbol{H}_{B,i}\|^2 + \sum_{j=1}^{q} \boldsymbol{\beta}_j^T \boldsymbol{S}_W \boldsymbol{\beta}_j, \\ &\text{subject to} \ \ \boldsymbol{A}^T \boldsymbol{A} = I \end{aligned} \tag{37}$$

where, $\boldsymbol{H}_{B,i}$ *is the* i*th row of* $\boldsymbol{H}_B$*. Then* $\hat{\boldsymbol{\beta}}_j, j = 1, 2, \ldots, q$*, span the same subspace as* $\boldsymbol{v}_j, j = 1, 2, \ldots, q$*.* [refer to [73] for the proof].

After establishing the equivalence, the regularization is applied on the *least squares* formulation in (37) via the $Lasso\text{-}penalty$ as shown below:

$$\begin{aligned} &\underset{\boldsymbol{A},\boldsymbol{B}}{\text{minimize}} \ \sum_{i=1}^{n} \|\boldsymbol{R}_W^{-T} \boldsymbol{H}_{B,i} - \boldsymbol{A}\boldsymbol{B}^T \boldsymbol{H}_{B,i}\|^2 + \sum_{j=1}^{q} \boldsymbol{\beta}_j^T \boldsymbol{S}_W \boldsymbol{\beta}_j, + \sum_{j=1}^{q} \lambda_{j,1} \|\boldsymbol{\beta}_j\|_1, \\ &\text{subject to} \ \ \boldsymbol{A}^T \boldsymbol{A} = I \end{aligned} \tag{38}$$

Since (38) is non-convex, finding the global optimum is often difficult. Qiao et al. [73], suggest a technique to obtain a local optimum by alternating optimization over $\boldsymbol{A}$ and $\boldsymbol{B}$. We refer readers to their article for details on their implementation.

Clemmensen et al. [18] also propose a similar sparse model using FLDA for classification problems. They also follow the approach of re-casting the optimization problem of FLDA into an equivalent least squares problem and then inducing sparsity by introducing a regularization term. However, the reformulation is achieved via an *optimal scoring* function that maps categorical variables to continuous variables via a sequence of *scorings*. Given a data matrix $\boldsymbol{X} \in \Re^{n\times p}$ and the samples belonging to one of the K classes, the equivalent regression problem can be formulated as:

$$\begin{aligned} \underset{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k}{\text{minimize}} \quad & ||\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\boldsymbol{\beta}_k||_2^2 \\ \text{subject to} \quad & \frac{1}{n}\boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y}\boldsymbol{\theta}_k = 1 \\ & \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y}\boldsymbol{\theta}_l = 0, \quad \forall l < k, \end{aligned} \tag{39}$$

where $\boldsymbol{\theta}_k$ is the score vector and $\boldsymbol{\beta}_k$ is the coefficient vector. It can be shown that the optimal vector $\boldsymbol{\beta}_k$ from (39) is also optimal to FLDA formulation in (28). Sparse discriminant vectors are then obtained by adding an l_1-penalty to the objective function in (39) as:

$$\begin{aligned} \underset{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k}{\text{minimize}} \quad & ||\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\boldsymbol{\beta}_k||_2^2 + \gamma \boldsymbol{\beta}_k^T \boldsymbol{\Omega} \boldsymbol{\beta}_k + \lambda ||\boldsymbol{\beta}_k||_1 \\ \text{subject to} \quad & \frac{1}{n}\boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y}\boldsymbol{\theta}_k = 1 \\ & \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y}\boldsymbol{\theta}_l = 0, \quad \forall l < k, \end{aligned} \tag{40}$$

where $\boldsymbol{\Omega}$ is a positive-definite matrix. The authors propose a simple iterative algorithm to obtain a local minima for the optimization problem in (40). The algorithm involves holding $\boldsymbol{\theta}_k$ fixed and optimizing with respect to $\boldsymbol{\beta}_k$, and holding $\boldsymbol{\beta}_k$ fixed and optimizing with respect to $\boldsymbol{\theta}_k$ until a pre-defined convergence criteria is met.

3.5 Discriminant Functions for High-Dimensional Data Classification

Linear discriminant analysis and QDA require the covariance within classes to be known a priori in order to establish a discriminant rule in classification problems. In many problems, since the covariance is not known a priori, researchers often attempt to estimate the covariance from the sample data. However, in high-dimensional problems, the sample covariance matrix is ill-conditioned and hence induces singularity in the estimation of the inverse covariance matrix. FLDA also faces similar challenges since *within-scatter* and *in-between scatter* are estimated from the sample data. In fact, even if the true covariance matrix is not ill-conditioned, the singularity of the sample covariance matrix will make these methods inapplicable when the dimensionality is larger than the sample size. Several authors performed a theoretical study on the performance of FLDA in high-dimensional classification settings. Bickel and Levina [2] showed that under some regularity conditions, as the ratio of features p and the number of samples n tend to infinity, the worst case misclassification rate tends to 0.5. This proves that as the dimensionality increases, FLDA is only as good as random guessing.

Several alternatives have been proposed to overcome the problem of singularity in LDA and QDA. Thomaz and Gillies [84] propose a new LDA algorithm (NLDA), which replaces the less reliable smaller eigenvalues of the sample covariance matrix with the grand mean of all eigenvalues and keeps larger eigenvalues unchanged. NLDA has been used successfully in face recognition problems. Xu et al. [94] state the lack of theoretical basis for NLDA and introduced a modified version of LDA called MLDA, which is based on a well-conditioned estimator for high-dimensional covariance matrices. This estimator has been shown to be more accurate than the sample covariance matrix asymptotically.

The assumption of independence in DLDA greatly reduces the number of parameters in the model and often results in an effective and interpretable classifier. Despite the fact that features will rarely be independent within a class, in the case of high-dimensional classification problems, the dependencies cannot be estimated due to lack of data. DLDA is shown to perform well for high-dimensional classification setting in spite of this naive assumption. Bickel and Levina [2] theoretically showed that it will outperform classical discriminant analysis in high-dimensional problems. However, one shortcoming of DLDA is that it uses all features and hence is not convenient for interpretation. Tibshirani et al. [86] introduced further regularization in DLDA using a procedure called *nearest shrunken centroids (NSC)* in order to improve misclassification error as well as interpretability. The regularization is introduced in a way that automatically assigns a weight *zero* to features that do not contribute to the class predictions. This is achieved by shrinking the classwise mean toward the overall mean, for each feature separately. We refer readers to [86] for a complete description of the method. DLDA integrated with *NSC* was applied to gene expression array analysis and is shown to be more accurate than other competing methods. The authors prove that the method is highly efficient in finding genes representative of small round blue cell tumors and leukemias. Several variations of NSC also exist in literature, for example [19,87]. Interestingly, NSC is also shown to be highly successful in open-set classification problems [77, 78] where the number of classes is not necessarily closed.

Another framework applied to high-dimensional classification problems include combining DLDA with shrinkage [71, 88]. Pang et al. [71] combined the shrinkage estimates of variances with diagonal discriminant scores to define two shrinkage-based discriminant rules called shrinkage-based DQDA (SDQDA) and shrinkage-based DLDA (SDLDA). Furthermore, the authors also applied regularization to further improve the performance of SDQDA and SDLDA. The discriminant rule combining shrinkage-based variances and regularization in diagonal discriminant analysis showed improvement over the original DQDA and DLDA, SVM, and k-Nearest Neighbors in many classification problems. Recently, Huang et al. [48] observed that the diagonal discriminant analysis suffers from serious drawback of having biased discriminant scores. Hence, they proposed bias-corrected diagonal discriminant rules by considering unbiased estimates for the discriminant scores. Especially in the case of highly unbalanced classification problems, the bias corrected rule is shown to outperform the standard rules.

Recently, SLDA has shown promise in high-dimensional classification problems. In [73], SLDA was applied to synthetic and real-world datasets including wine datasets and gene expression datasets and is shown to perform very well on training and testing data with lesser number of significant variables. The authors in [18] compared SLDA obtained via optimal scoring to other methods like shrunken centroid regularized discriminant analysis, sparse partial least squares regression and the elastic-net regression on a number of high-dimensional datasets and is shown to have comparable performance to other methods but with lesser number of significant variables.

4 Hybrid Classifiers

We now discuss an important set of classifiers that are frequently used for classification in the context of high-dimensional data problems. High dimensional datasets usually consist of irrelevant and redundant features that adversely effect the performance of traditional classifiers. Also, the high dimensionality of the data makes the estimation of statistical measures difficult. Hence, several techniques have been proposed in the literature to perform feature selection that selects relevant features suitable for classification [46]. Generally, feature selection is performed as a dimensionality reduction step prior to building the classification model using the traditional classifiers. Unlike other dimensionality reduction techniques like those based on transformation (e.g., principal component analysis) or compression (e.g., based on information theory), feature selection techniques do not alter the original dimensional space of the features, but merely select a subset of them [76]. Thus, they offer the advantage of interpretability by a domain expert as they preserve the original feature space. Also, feature selection helps to gain a deeper insight into the underlying processes that generated the data and thus plays a vital role in the discovery of *biomarkers* especially in biomedical applications [30]. Thus the classification framework can be viewed as a two-stage process with dimensionality reduction via feature selection being the first step followed by a classification model. We call these set of classifiers as *hybrid classifiers*, as different techniques pertaining to two stages have been combined to produce classification frameworks that have been successful in several high-dimensional problems. We briefly describe various feature selection techniques and also review the hybrid classifiers developed using these techniques for high-dimensional data problems.

4.1 Feature Selection Methods

Recently, feature selection has been an active area of research among many researchers due to tremendous advances in technology enabling collecting samples with hundreds and thousands of attributes in a single experiment. The goal of

feature selection techniques is to find an *optimal* set of features based on different measures of *optimality*. Irrespective of the measure of optimality, the selected subset of features should ideally possess the following characteristics [40]:

- The cardinality of the subset should be minimal such that it is necessary and sufficient to accurately predict the class of unknown samples,
- The subset of features should improve the prediction accuracy of the classifier run on data containing only these features rather than on the original dataset with all the features,
- The resulting class distribution, given only the values for the selected features, is as close as possible to the original class distribution given all feature values.

Based on the above feature characteristics, it is obvious that *irrelevant* features would not be part of the optimal set of features, where an irrelevant feature with respect to the target class is defined as follows [97].

Let F be the full set of features and C be the target class. Define $F_i \in F$ and $S_i = F - F_i$.

Definition 1 (Irrelevance). A feature F_i is irrelevant if and only if

$$\forall S_i^{'} \subseteq S_i, \quad \mathbf{P}(C\,|\,F_i, S_i^{'}) = \mathbf{P}(C\,|\,S_i^{'})$$

Irrelevance simply means that it is not necessary for classification since the class distribution given any subset of other features does not change after eliminating the feature.

The definition of relevance is not as straightforward as irrelevance. There have been several definitions for relevance in the past; however, Kohavi and John [58] argued that the earlier definitions weren't adequate to accurately classify the features. Hence, they defined relevance in terms of an optimal Bayes classifier. A feature F_i is strongly relevant if removal of F_i alone will result in decrease of performance of an optimal Bayes classifier. A feature F_i is weakly relevant if it is not strongly relevant and there exists a subset of features, $S_i^{'}$, such that the performance of a Bayes classifier on $S_i^{'}$ is worse than the performance on $S_i^{'}$ U $\{F_i\}$.

Definition 2 (Strong Relevance). A feature F_i is strongly relevant if only and if:

$$\mathbf{P}(C\,|\,F_i, S_i^{'}) \neq \mathbf{P}(C\,|\,S_i^{'}), \quad S_i^{'} \subseteq S_i \tag{41}$$

Definition 3 (Weak Relevance). A feature F_i is weakly relevant if only and if:

$$\mathbf{P}(C\,|\,F_i, S_i) = \mathbf{P}(C\,|\,S_i) \quad \text{and,} \quad \exists S_i^{'} \subseteq S_i, \quad \mathbf{P}(C\,|\,F_i, S_i^{'}) \neq \mathbf{P}(C\,|\,S_i^{'}) \tag{42}$$

Strong relevance implies that the feature is indispensable and is required for an optimal set, while weak relevance implies that the feature may be required sometimes to improve the prediction accuracy. From this, one may conclude that the optimal set should consist of all the strongly relevant features, none of the irrelevant

features and some of the weakly irrelevant features. However, the definitions do not explicitly mention which of the weakly relevant features should be included and which of them excluded. Hence, Yu and Liu [97] claim that the weakly relevant features should be further classified to discriminate among the redundant features and the non-redundant features, since earlier research efforts showed that along with irrelevant features, redundant features also adversely affect the classifier performance. Before we provide definitions, we introduce another concept called feature's *Markov Blanket* as defined by Koller and Sahami [59].

Definition 4 (Markov Blanket). Given a feature F_i, let $M_i \subset F(F_i \notin M_i)$, M_i is said to be a Markov blanket for F_i if only and if:

$$\mathbf{P}(F - M_i - \{F_i\}, C | F_i, M_i) = P(F - M_i - \{F_i\}, C | M_i) \tag{43}$$

The Markov blanket M_i could be imagined as a *blanket* for the feature F_i that subsumes not only the information that F_i possesses about target class C, but also about other features. It is also important to note that the strongly relevant features cannot have a Markov Blanket. Since the irrelevant features do not contribute to classification, Yu and Liu [97] further classified the weakly relevant features into either *redundant* or *non-redundant* using the concept of Markov blanket:

Definition 5 (Redundant Feature). Given a set of current features G, a feature is redundant and hence should be removed from G if and only if it has a Markov Blanket within G.

From the above definitions, it is clear that the optimal set of features should consist of all of the strongly relevant features and the weakly relevant non-redundant features. However, an exhaustive search over the feature space is intractable since there are 2^p possibilities with p being the number of features. Hence, over the past decade, several heuristic and approximate methods have been developed to perform feature selection. In the context of classification, feature selection techniques can be organized into three categories, depending on how they combine the feature selection search with the construction of the classification model: filter methods, wrapper methods and embedded methods [76]. While all methods define some criterion measure to eliminate the irrelevant features, very few methods attempt to eliminate the redundant features as well. Here, we briefly describe methods in each of the three categories.

4.1.1 Filter Methods

Filter methods assess feature relevance from the intrinsic properties of the data. In most cases the features are ranked using a feature relevance score and the low-scoring features are removed. The reduced data obtained from considering only the selected features are then presented as an input to the classification algorithm. Filter techniques offer several advantages including scalability to high-dimensional

datasets, being computationally efficient, and are independent of the classification algorithm. This independency offers the advantage of performing feature selection only once and then evaluating different classifiers.

Some univariate filter techniques perform simple hypothesis testing like Chi-Square (χ^2) test or t-test to eliminate the irrelevant features, while other techniques estimate information theoretic measures like information gain and gain-ratio to perform the filtering process [1]. Although these techniques are simple, fast and highly scalable, they ignore feature dependencies which may lead to worse classification performance as compared with other feature selection techniques. In order to account for feature dependencies, a number of multivariate filter techniques were introduced. The multivariate filter methods range from accounting for simple mutual interactions [4] to more advanced solutions exploring higher order interactions. One such technique called correlation-based feature selection (CFS) introduced by Hall [42], evaluates a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them:

$$\mathrm{CFS}_S = \frac{k\Phi_{cf}}{\sqrt{k + k(k-1)\Phi_{ff}}} \tag{44}$$

where CFS_S is the score of a feature subset S containing k features, Φ_{cf} is the average feature-to-class correlation ($f \in S$), and Φ_{ff} is the average feature-to-feature correlation. Unlike the univariate filter methods, CFS presents a score for a subset of features. Since, exhaustive search is intractable, several heuristic techniques like greedy hill-climbing or best-first search have been proposed to find the feature subset with the highest CFS score.

Another important multivariate filter method called Markov blanket filtering was introduced by Koller and Sahami [59]. The idea here being that once we find a Markov blanket of feature F_i in a feature set G, we can safely remove F_i from G without compromising on the class distribution. Since estimating the Markov blanket for a feature is hard, Koller and Sahami propose a simple iterative algorithm that starts with the full feature set $F = G$ and then repeatedly eliminates one feature at a time based on cross-entropy of each feature until a pre-selected number of features are removed.

Koller and Sahami further prove that in such a sequential elimination process in which unnecessary features are removed one by one, a feature tagged as unnecessary based on the existence of a Markov blanket M_i remains unnecessary in later stages when more features have been removed. Also, the authors claim that the process removes all the irrelevant as well as redundant features. Several variations to the Markov blanket filtering method like Grow-Shrink (GS) algorithms, incremental association Markov blanket (IAMB), Fast-IAMB and recently λ-IAMB have been proposed by other authors [36]. Due to space constraints, we mention other interesting multivariate filter methods like fast-correlation-based feature selection (FCBF) ([96]), minimum redundancy-maximum relevance (MRMR) [26], and uncorrelated shrunken centroid (USC) [95] algorithms.

Statnikov et al. [82] recently performed a comprehensive comparative study between Random Forests [8] and SVM for microarray-based cancer classification. They adopt several filter methods like sequential filtering techniques as a pre-processing step to select a subset of features which are then used as input to the classifiers. It is shown that on an average, SVM outperform Random Forests on most microarray datasets. Recently, Pal and Moody [68] studied the effect of dimensionality on performance of SVM using four feature selection techniques namely CFS, MRMR, Random Forests and SVM-RFE [41] on hyperspectral data. Unlike earlier findings, they show that dimensionality might affect the performance of SVM and hence a pre-processing step like feature selection might still be useful to improve the performance.

4.1.2 Wrapper Methods

As seen in the earlier section, filter methods treat the problem of finding a good feature subset independently of the classifier building step. Wrapper methods, on the other hand, integrate the classifier hypothesis search within the feature subset search. In this framework, a search procedure in the feature space is first defined, and various subsets of features are generated and evaluated. The evaluation of a specific feature subset is obtained by training and testing a specific classification model, making this approach tailored to a specific classification algorithm [58, 76]. Advantages of wrapper methods include consideration of feature dependencies and the ability to include interactions between the feature subset search and model selection. A common drawback includes the risk of higher overfitting than the filter methods and could be computationally intensive if the classification model especially has a high computational cost.

The wrapper methods generally employ a search algorithm in order to search through the space of all feature subsets. The search algorithm is *wrapped* around the classification model which provides a feature subset that can be evaluated by the classification algorithm. As mentioned earlier, since an exhaustive search is not practical, heuristic search methods are used to guide the search. These search methods can be broadly classified as deterministic and randomized search algorithms. Deterministic search methods include a set of sequential search techniques like the Sequential Forward Selection [56], Sequential Backward Selection [56], Plus-l Minus-r Selection [31], Bidirectional Search, Sequential Floating Selection [72] etc., where the features are either sequentially added or removed based on some criterion measure. Randomized Search algorithms include popular techniques like Genetic Algorithms [20], Simulated Annealing [55], Randomized Hill Climbing [81], etc.

4.1.3 Embedded Methods

Embedded methods integrate the search for an optimal subset of features into the classifier construction and can be seen as a search in the combined space of feature subsets and hypotheses. Similar to wrapper methods, embedded approaches are also specific to a given learning algorithm. The advantages of embedded methods include the interaction with the classification model, but unlike the wrapper methods, also has the advantage to be less computationally intensive [76].

Recently embedded methods have gained importance among the research community due to their advantages. The embedded characteristic of several classifiers to eliminate input features futile to classification and thus select a subset of features, has been exploited by several authors. Examples include the use of random forests (discussed later) in an embedded way to calculate the importance of each feature [24, 51]. Another line of embedded feature selection techniques uses the weights of each feature in linear classifiers, such as SVM [41] and logistic regression [63]. These weights are used as a measure of relevance of each feature, and thus allow for the removal of features with very small weights. Also, recently regularized classifiers like Lasso and elastic-net have also been successfully employed in performing feature selection in microarray gene analysis [99]. Another interesting technique called feature selection via sparse SVM has been recently proposed by Tan et al. [83]. This technique called the feature Generating machine (FGM) adds a binary variable for every feature in the sparse formulation of SVM via l_0-norm and the authors propose a cutting plane algorithm combined with multiple kernel learning to efficiently solve the convex relaxation of the optimization problem.

5 Ensemble Classifiers

Ensemble classifiers have gained increasing attention from the research community over the past years, ranging from simple averaging of individually trained neural networks to the combination of thousands of decision trees to build Random Forests [8], to the boosting of weak classifiers to build a strong classifier where the training of each subsequent classifier depends on the results of all previously trained classifiers [75]. The main idea of an ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions. They combine multiple hypotheses of different models with the hope to form a better classifier. Alternatively, an ensemble classifier can also be viewed as a technique for combining many weak learners in an attempt to produce a strong learner. Hence an ensemble classifier is itself a supervised learning algorithm capable of making prediction on unknown sample data. The trained ensemble classifier, therefore, represents a single hypothesis that is not necessarily contained within the hypothesis space of the constituent models. This flexibility of ensemble classifiers can theoretically overfit to the training data more than a single model

would, but however surprisingly, in practice, some ensemble techniques (especially bagging and Random Forests) tend to reduce problems related to overfitting of the training data.

In the past few years, experimental studies show that combining the outputs of multiple classifiers similar to ensemble methods reduces the generalization error [25]. Ensemble methods are particularly effective due to the phenomenon that various types of classifiers have different inductive biases. Additionally, ensemble methods can effectively make use of such diversity to reduce the variance-error while keeping the bias-error in check. In certain situations, an ensemble can also reduce bias-error, as shown by the theory of large margin classifiers. So, diversified classifiers help in building a lesser number of classifiers, especially in the case of Random Forests. The increase in prediction accuracy does come at a cost of performing more calculations in comparison to a single model. So, the ensemble methods can be thought of as a way to compensate for a poor learner by performing a lot of computations. So, a fast poor learner like decision trees have certainly gained from ensemble methods; although slow algorithms can also benefit from ensemble techniques.

Recently, ensemble methods have shown promise in high-dimensional data classification problems. In particular, bagging methods, random forests and boosting have been particularly impressive due to their flexibility to create stronger classifiers from weak classifiers. Here, we describe two methods: AdaBoost and Random Forests, and show their importance in high-dimensional problems.

5.1 AdaBoost

Boosting [33, 79] is a general method which attempts to boost the accuracy of any given learning algorithm. The inception of boosting can be traced back to a theoretical framework for studying machine learning called the "PAC" learning model, [91]. Kearns and Valiant [54] were among the first authors to pose the question of whether a *weak* learner which is only slightly correlated with the true classification and performs just slightly better than random guessing in the PAC model can be *boosted* into an accurate *strong* learning algorithm that is arbitrarily well-correlated with true classification. Schapire [79] proposed the first provable polynomial-time boosting algorithm in 1989. A year later, Freund [32] developed a much more efficient boosting algorithm which, although optimal in a certain sense, nevertheless suffered from certain practical drawbacks.

Boosting encompasses a family of methods that produces a series of classifiers. The training set used for each member of the series is chosen based on the performance of the earlier classifier(s) in the series. Unlike other committee methods like bagging [6], in boosting, the base classifiers are trained in sequence, and each base classifier is trained using a weighted variant of the dataset in which the individual weighting coefficient depends on the performance of previous classifiers.

In particular, points that are misclassified by one of the base classifiers are given greater weight when used to train the next classifier in the sequence. Once all the classifiers have been trained, their predictions are then combined through a weighted majority voting scheme.

AdaBoost, short for Adaptive Boosting, formulated by Yoav Freund and Robert Schapire [33], solved many of the practical difficulties of earlier boosting algorithms. It can be considered as classification framework that can be used in conjunction with many other learners to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. The framework provides a new weak classifier with a form of training set that is representative of the performance of previous classifiers. The weights of those training samples that are misclassified by earlier weak learners are given higher values than those that are correctly classified. This allows the new classifier to adapt to the misclassified training samples and focus on predicting them correctly. After the training phase is complete, each classifier is assigned a weight and their outputs are linearly combined to make predictions on the unknown sample. Generally, it provides a significant performance boost to weak learners that are only slightly better than random guessing. Even classifiers with a higher error rate could also be useful as they will have negative coefficients in the final linear combination of classifiers and hence behave like their inverses.The precise form of the AdaBoost algorithm is described below.

Consider a binary classification problem, in which the training data comprises input vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ along with corresponding binary target variables given by t where $t_n \in \{-1, 1\}$. Each data point is given an associated weighting parameter w_n, which is initially set 1/N for all data points. We assume that we have a procedure available for training a base classifier using weighted data to give a function $y(\boldsymbol{x}) \in \{-1, 1\}$.

- Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, 2, \ldots, N$.
- For $m = 1, \ldots, M$:
  (a) Fit a classifier $y_m(\boldsymbol{x})$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n) \tag{45}$$

  where $I(y_m(\boldsymbol{x}_n) \neq t_n)$ is the indicator function and equals 1 when $(y_m(\boldsymbol{x}_n) \neq t_n)$ and 0 otherwise.
  (b) Evaluate the quantities

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\boldsymbol{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \tag{46}$$

and then use ϵ_m to evaluate

$$\alpha_m = \ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\} \tag{47}$$

(c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\boldsymbol{x}_n) \neq t_n)\} \tag{48}$$

- Make predictions using the final model, which is given by:

$$Y_M^{(\boldsymbol{x})} = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x})\right) \tag{49}$$

We see that the first weak learner $y_1(\boldsymbol{x})$ is trained using weighting coefficients $w_n^{(1)}$ that are all equal and hence is similar to training a single classifier. From (48), we see that in subsequent iterations the weighting coefficients $w_n^{(m)}$ are increased for data points that are misclassified and decreased for data points that are correctly classified. Successive classifiers are therefore forced to focus on points that have been misclassified by previous classifiers, and data points that continue to be misclassified by successive classifiers receive even greater weight. The quantities ϵ_m represent weighted measures of the error rates of each of the base classifiers on the dataset. We therefore see that the weighting coefficients α_m defined by (47) give greater weight to more accurate classifiers when computing the overall output for unknown samples given by (49). AdaBoost is sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than most learning algorithms. We refer readers to [34] for a more theoretical discussion on the performance of the AdaBoost algorithm.

Boosting framework in conjunction with several classifiers have been successfully applied to high-dimensional data problems. As discussed in [7] boosting framework can be viewed as a functional gradient descent technique. This analysis of boosting connects the method to more common optimization view of statistical inference. Bühlmann and Yu [11] investigate one such computationally simple variant of boosting called L_2Boost, which is constructed from a functional gradient descent algorithm with the L_2-loss function. In particular, they study the algorithm with cubic smoothing spline as the base learner and show empirically on real and simulation datasets the effectiveness of the algorithm in high-dimensional predictors. Bühlmann [10] presented an interesting review on how the boosting methods can be useful for high-dimensional problems. He proposes that inherent variable selection and assigning variable amount of degrees of freedom to the selected variables by boosting algorithms could be a reason for high performance in high-dimensional problems. Additionally, he suggests that boosting yields consistent function approximations even when the number of predictors grow fast to infinity,

where the underlying true function is *sparse*. Dettling and Bühlmann [23] applied boosting to perform classification tasks with gene expression data. A modified boosting framework in conjunction with decision trees that does pre-selection was proposed and shown to yield slight to drastic improvement in performance on several publicly available datasets.

5.2 Random Forests

Random forests are an ensemble classifier that consists of many tree-type classifiers with each classifier being trained on a bootstrapped sample of the original training data, and searches only across a randomly selected subset of the input variables to determine a split (for each node). For classification, each tree in the Random Forest casts a unit vote for the most popular class at input x. The output of the Random Forest for an unknown sample is then determined by a majority vote of the trees. The algorithm for inducing Random Forests was developed by Leo Breiman [8] and can be summarized as below:

Assume the number of training samples be N, and the number of features be given by M. Also, assume that random m number of features ($m < M$) used for decision at each split. Each tree in the Random Forest is constructed as follows:

- Choose a training set for this tree by bootstrapping the original training set n times. The rest of the samples are used as a testing set to estimate the error of the tree.
- For each node of the tree, the best split is based on randomly choosing m features for each training sample and the tree is fully grown without pruning.

For prediction, a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the class obtained from majority vote of all the trees is reported as Random Forest prediction.

Random Forests are considered one of the most accurate classifiers and are reported to have several advantages. Random Forests are shown to handle many features and also assign a weight relative to their importance in classification tasks which can be further explored for feature selection. The computational complexity of the algorithm is reduced as the number of features used for each split is bounded by m. Also, non-pruning of the trees also helps in reducing the computational complexity further. Such random selection of features to build the trees also limits the correlation among the trees thus resulting in error rates similar to those of AdaBoost. The analysis of Random Forests shows that its computational time is $cT\sqrt{M}N\ \log(N)$ where c is a constant, T is the number of trees in the ensemble, M is the number of features and N is the number of training samples in the dataset. It should be noted that although Random Forests are not computationally intensive, they require a fair amount of memory as they store an N by T matrix in memory. Also, Random Forests have sometimes been shown to overfit to the data in some classification problems.

Random Forests, due to the aforementioned advantages, can handle high-dimensional data by building a large number of trees using only a subset of features. This combined with the fact that the random selection of features for a split seeks to minimize the correlation between the trees in the ensemble, certainly helps in building an ensemble classifier with high generalization accuracy for high-dimensional data problems. Gislason et al. [38] performed a comparative study among Random Forests and other well-known ensemble methods for multisource remote sensing and geographic data. They show that Random Forests outperform a single CART classifier and perform on par with other ensemble methods like bagging and boosting. On a related remote sensing application, Pal [67] investigated the use of Random Forests for classification tasks and compared their performance with SVM. Pal showed that Random Forests perform equally well to SVM in terms of classification accuracy and training time. Additionally, Pal concludes that the user-defined parameters in Random Forests are less than those required for SVM. Pang et al. [70] proposed a pathway-based classification and regression method using Random Forests to analyze gene expression data. The proposed method allows to rank important pathways, discover important genes and find pathway-based outlying cases. Random Forests, in comparison with other machine learning algorithms, were shown to have either lower or second-lowest classification error rates. Recently, Genuer et al. [37] used Random Forests to perform feature selection as well. The authors propose a strategy involving ranking of the explanatory variables using the Random Forests score of importance.

6 Software Packages

We briefly describe some publicly available resources that have implemented the methods discussed here. These packages are available in several programming languages including Java, Matlab and R softwares. LibSVM [14] is an integrated software that implements SVM and offers several extensions to Java, C++, Python, R and Matlab. Weka [43] is a collection of machine learning algorithms for data mining tasks implemented in Java. It contains methods to perform classification as well as feature selection on high-dimensional datasets. The FSelector package in R language offers several algorithms to perform filter, wrapper and embedded feature selection. Several packages are also available to perform regularization. Glmnet for Matlab[1] solves for regularized paths in Generalized Linear models while Lasso2[2] and LARS[3] packages provide similar algorithms in R language. Random Forests and AdaBoost algorithms are also available via randomForest[4] and adabag[5] packages in R language.

[1] http://www-stat.stanford.edu/~tibs/glmnet-matlab/.

[2] http://cran.r-project.org/web/packages/lasso2/index.html.

[3] http://cran.r-project.org/web/packages/lars/index.html.

[4] http://cran.r-project.org/web/packages/randomForest/index.html.

[5] http://cran.r-project.org/web/packages/adabag/index.html.

7 Concluding Remarks

We have presented several classification problems for high-dimensional data problems. Several researchers have focused on extended the traditional algorithms like LDA and Logistic Regression in the context of high-dimensional data settings. Though some success is seen on this front, recently, the focus has shifted to applying regularization techniques and ensemble type methods to make more accurate predictions. Though the progress made so far is encouraging, we believe that high-dimensional data classification would continue to be an active area of research as the technological innovations continue to evolve and become more effective.

Acknowledgements This research is partially supported by NSF & DTRA grants

References

1. Ben-Bassat, M.: 35 use of distance measures, information measures and error bounds in feature evaluation. In: Handbook of Statistics, vol. 2, pp. 773–791. North-Holland, Amsterdam (1982)
2. Bickel, P., Levina, E.: Some theory for fisher's linear discriminant function, Naive Bayes', and some alternatives when there are many more variables than observations. Bernoulli **10**(6), 989–1010 (2004)
3. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006)
4. Bo, T., Jonassen, I.: New feature subset selection procedures for classification of expression profiles. Genome Biol. **3**(4), 1–11 (2002)
5. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
6. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
7. Breiman, L.: Prediction games and arcing algorithms. Neural Comput. **11**(7), 1493–1517 (1999)
8. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
9. Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M., Haussler, D.: Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl. Acad. Sci. USA **97**(1), 262 (2000)
10. Bühlmann, P.: Boosting methods: why they can be useful for high-dimensional data. In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC) (2003)
11. Bühlmann, P., Yu, B.: Boosting with the l 2 loss: regression and classification. J. Am. Stat. Assoc. **98**(462), 324–339 (2003)
12. Burges, C.: Advances in Kernel Methods: Support Vector Learning. The MIT Press, Cambridge (1999)
13. Byvatov, E., Schneider, G., et al.: Support vector machine applications in bioinformatics. Appl. Bioinformatics **2**(2), 67–77 (2003)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011)
15. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Mach. Learn. **46**(1), 131–159 (2002)
16. Chung, K., Kao, W., Sun, C., Wang, L., Lin, C.: Radius margin bounds for support vector machines with the rbf kernel. Neural Comput. **15**(11), 2643–2681 (2003)

17. Clarke, R., Ressom, H., Wang, A., Xuan, J., Liu, M., Gehan, E., Wang, Y.: The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nat. Rev. Cancer **8**(1), 37–49 (2008)
18. Clemmensen, L., Hastie, T., Witten, D., Ersbøll, B.: Sparse discriminant analysis. Technometrics **53**(4), 406–413 (2011)
19. Dabney, A.: Classification of microarrays to nearest centroids. Bioinformatics **21**(22), 4148–4154 (2005)
20. Davis, L., Mitchell, M.: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
21. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.: The mahalanobis distance. Chemometr. Intell. Lab. Syst. **50**(1), 1–18 (2000)
22. Den Hertog, D.: Interior Point Approach to Linear, Quadratic and Convex Programming: Algorithms and Complexity. Kluwer Academic, Norwell (1992)
23. Dettling, M., Bühlmann, P.: Boosting for tumor classification with gene expression data. Bioinformatics **19**(9), 1061–1069 (2003)
24. Díaz-Uriarte, R., De Andres, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics **7**(3), 1–13 (2006)
25. Dietterich, T.: Ensemble methods in machine learning. In: Multiple Classifier Systems, pp. 1–15. Springer, Heidelberg (2000)
26. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J. Bioinforma. Comput. Biol. **3**(2), 185–205 (2005)
27. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley-Interscience, London (2001)
28. Dudoit, S., Fridlyand, J., Speed, T.: Comparison of discrimination methods for the classification of tumors using gene expression data. J. Am. Stat. Assoc. **97**(457), 77–87 (2002)
29. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)
30. Fenn, M., Pappu, V.: Data mining for cancer biomarkers with raman spectroscopy. In: Data Mining for Biomarker Discovery, pp. 143–168. Springer, Berlin (2012)
31. Ferri, F., Pudil, P., Hatef, M., Kittler, J.: Comparative study of techniques for large-scale feature selection. In: Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies, and Hybrid Systems, pp. 403–413. IEEE Xplore (1994)
32. Freund, Y.: Boosting a weak learning algorithm by majority. Inf. Comput. **121**(2), 256–285 (1995)
33. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, pp. 148–156. Morgan Kaufmann, Los Altos (1996)
34. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(1612), 771–780 (1999)
35. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. Springer Series in Statistics. Springer, Berlin (2001)
36. Fu, S., Desmarais, M.: Markov blanket based feature selection: a review of past decade. In: Proceedings of the World Congress on Engineering, vol. 1, pp. 321–328 (2010). Citeseer
37. Genuer, R., Poggi, J., Tuleau-Malot, C.: Variable selection using random forests. Pattern Recognit. Lett. **31**(14), 2225–2236 (2010)
38. Gislason, P., Benediktsson, J., Sveinsson, J.: Random forests for land cover classification. Pattern Recognit. Lett. **27**(4), 294–300 (2006)
39. Guo, X., Yang, J., Wu, C., Wang, C., Liang, Y.: A novel ls-svms hyper-parameter selection based on particle swarm optimization. Neurocomputing **71**(16), 3211–3215 (2008)
40. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
41. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1), 389–422 (2002)
42. Hall, M.: Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato (1999)

43. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newslett. **11**(1), 10–18 (2009)
44. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Englewood (2004)
45. Herbert, P., Tiejun, T.: Recent advances in discriminant analysis for high-dimensional data classification. J. Biom. Biostat. **3**(2), 1–2 (2012)
46. Hua, J., Tembe, W., Dougherty, E.: Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognit. **42**(3), 409–424 (2009)
47. Huang, C., Wang, C.: A ga-based feature selection and parameters optimization for support vector machines. Expert Syst. Appl. **31**(2), 231–240 (2006)
48. Huang, S., Tong, T., Zhao, H.: Bias-corrected diagonal discriminant rules for high-dimensional classification. Biometrics **66**(4), 1096–1106 (2010)
49. Hughes, G.: On the mean accuracy of statistical pattern recognizers. IEEE Trans. Inf. Theory **14**(1), 55–63 (1968)
50. Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000)
51. Jiang, H., Deng, Y., Chen, H., Tao, L., Sha, Q., Chen, J., Tsai, C., Zhang, S.: Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics **5**(81), 1–12 (2004)
52. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Machine Learning: ECML-98, pp. 137–142. Springer, Berlin (1998)
53. Johnstone, I., Titterington, D.: Statistical challenges of high-dimensional data. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **367**(1906), 4237–4253 (2009)
54. Kearns, M., Valiant, L.: Learning Boolean formulae or finite automata is as hard as factoring. Center for Research in Computing Technology, Aiken Computation Laboratory, Harvard University (1988)
55. Kirkpatrick, S., Gelatt, C. Jr., Vecchi, M.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
56. Kittler, J.: Feature set search algorithms. In: Pattern Recognition and Signal Processing, pp. 41–60. Sijthoff and Noordhoff, Alphen aan den Rijn (1978)
57. Kleinbaum, D., Klein, M., Pryor, E.: Logistic Regression: A Self-learning Text. Springer, Berlin (2002)
58. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif. Intell. **97**(1–2), 273–324 (1997)
59. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proceedings of the 13th International Conference on Machine Learning, pp. 284–292 (1996)
60. Köppen, M.: The curse of dimensionality. In: Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), pp. 4–8 (2000)
61. Lin, S., Lee, Z., Chen, S., Tseng, T.: Parameter determination of support vector machine and feature selection using simulated annealing approach. Appl. Soft Comput. **8**(4), 1505–1512 (2008)
62. Lin, S., Ying, K., Chen, S., Lee, Z.: Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst. Appl. **35**(4), 1817–1824 (2008)
63. Ma, S., Huang, J.: Regularized roc method for disease classification and biomarker selection with microarray data. Bioinformatics **21**(24), 4356–4362 (2005)
64. McLachlan, G., Wiley, J.: Discriminant Analysis and Statistical Pattern Recognition. Wiley Online Library, New York (1992)
65. Minh, H., Niyogi, P., Yao, Y.: Mercer's theorem, feature maps, and smoothing. In: Learning Theory, pp. 154–168. Springer Berlin Heidelberg (2006)
66. Mourão-Miranda, J., Bokde, A., Born, C., Hampel, H., Stetter, M.: Classifying brain states and determining the discriminating activation patterns: support vector machine on functional MRI data. NeuroImage **28**(4), 980–995 (2005)
67. Pal, M.: Support vector machine-based feature selection for land cover classification: a case study with dais hyperspectral data. Int. J. Remote Sens. **27**(14), 2877–2894 (2006)

68. Pal, M., Foody, G.: Feature selection for classification of hyperspectral data by svm. IEEE Trans. Geosci. Remote Sens. **48**(5), 2297–2307 (2010)
69. Pal, M., Mather, P.: Support vector machines for classification in remote sensing. Int. J. Remote Sens. **26**(5), 1007–1011 (2005)
70. Pang, H., Lin, A., Holford, M., Enerson, B., Lu, B., Lawton, M., Floyd, E., Zhao, H.: Pathway analysis using random forests classification and regression. Bioinformatics **22**(16), 2028–2036 (2006)
71. Pang, H., Tong, T., Zhao, H.: Shrinkage-based diagonal discriminant analysis and its applications in high-dimensional data. Biometrics **65**(4), 1021–1029 (2009)
72. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit. Lett. **15**(11), 1119–1125 (1994)
73. Qiao, Z., Zhou, L., Huang, J.: Sparse linear discriminant analysis with applications to high dimensional low sample size data. Int. J. Appl. Math. **39**(1), 6–29 (2009)
74. Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J., et al.: Multiclass cancer diagnosis using tumor gene expression signatures. Proc. Natl. Acad. Sci. USA **98**(26), 15149–15154 (2001)
75. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**(1), 1–39 (2010)
76. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007)
77. Schaalje, G., Fields, P.: Open-set nearest shrunken centroid classification. Commun. Stat. Theory Methods **41**(4), 638–652 (2012)
78. Schaalje, G., Fields, P., Roper, M., Snow, G.: Extended nearest shrunken centroid classification: a new method for open-set authorship attribution of texts of varying sizes. Lit. Linguist. Comput. **26**(1), 71–88 (2011)
79. Schapire, R.: The strength of weak learnability. Mach. Learn. **5**(2), 197–227 (1990)
80. Schoonover, J., Marx, R., Zhang, S.: Multivariate curve resolution in the analysis of vibrational spectroscopy data files. Appl. Spectrosc. **57**(5), 483–490 (2003)
81. Skalak, D.: Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Proceedings of the 11th International Conference on Machine Learning, pp. 293–301 (1994). Citeseer
82. Statnikov, A., Wang, L., Aliferis, C.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics **9**(319), 1–10 (2008)
83. Tan, M., Wang, L., Tsang, I.: Learning sparse svm for feature selection on very high dimensional datasets. In: Proceedings of the 27th International Conference on Machine Learning, pp. 1047–1054 (2010)
84. Thomaz, C., Gillies, D.: A maximum uncertainty lda-based approach for limited sample size problems - with application to face recognition. In: Proceedings of the 18th Brazilian Symposium on Computer Graphics and Image Processing, pp. 89–96. IEEE, Natal (2005)
85. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Series B Methodol. **58**, 267–288 (1996)
86. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc. Natl. Acad. Sci. **99**(10), 6567–6572 (2002)
87. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Class prediction by nearest shrunken centroids, with applications to dna microarrays. Stat. Sci. **18**, 104–117 (2003)
88. Tong, T., Chen, L., Zhao, H.: Improved mean estimation and its application to diagonal discriminant analysis. Bioinformatics **28**(4), 531–537 (2012)
89. Trafalis, T., Ince, H.: Support vector machine for regression and applications to financial forecasting. In: Proceedings of the International Joint Conference on Neural Networks, vol. 6, pp. 348–353. IEEE, New York (2000)
90. Trunk, G.: A problem of dimensionality: a simple example. IEEE Trans. Pattern Anal. Mach. Intell. **3**(3), 306–307 (1979)
91. Valiant, L.: A theory of the learnable. Commun. ACM **27**(11), 1134–1142 (1984)
92. Vapnik, V.: The nature of statistical learning theory. springer (2000)

93. Vapnik, V., Chapelle, O.: Bounds on error expectation for support vector machines. Neural Comput. **12**(9), 2013–2036 (2000)
94. Xu, P., Brock, G., Parrish, R.: Modified linear discriminant analysis approaches for classification of high-dimensional microarray data. Comput. Stat. Data Anal. **53**(5), 1674–1687 (2009)
95. Yeung, K., Bumgarner, R., et al.: Multiclass classification of microarray data with repeated measurements: application to cancer. Genome Biol. **4**(12), R83 (2003)
96. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th International Conference on Machine Learning, pp. 856–863 (2003)
97. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. **5**, 1205–1224 (2004)
98. Zhang, L., Lin, X.: Some considerations of classification for high dimension low-sample size data. Stat. Methods Med. Res. **22**, 537–550 (2011)
99. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B Stat. Methodol. **67**(2), 301–320 (2005)

Algorithm FRiS-TDR for Generalized Classification of the Labeled, Semi-labeled and Unlabeled Datasets

I.A. Borisova and N.G. Zagoruiko

Abstract The problem of generalized classification combines three well-known problems of machine learning: classification (supervised learning), clustering (unsupervised learning), and semi-supervised learning. These problems differ from each other based on the ratio of labeled and unlabeled objects in a training dataset. In the classification problem all the objects are labeled, and in the clustering problem all the objects are unlabeled. Semi-supervised learning makes use of both labeled and unlabeled objects for training—typically a small amount of labeled objects with a large amount of unlabeled objects. Usually these problems are examined separately and special algorithms are developed for solving each of them. Algorithm FRiS-taxonomy decision rule based on function of rival similarity examines these three problems as special cases of the generalized classification problem and solves all of them. This algorithm automatically determines the number of clusters and finds effective decision rules independently of the ratio of labeled and unlabeled samples in datasets.

Keywords FRiS-function • Semi-supervised learning • Clustering • Classification • Generalized classification

1 Introduction

The clustering and recognition (classification) are considered as close, but different, problems [1]. In the clustering problem, a set of unlabeled objects $V_u =< a_1, a_2, \ldots a_{Mu} >$ is given as input. It is supplemented with hypotheses that the training dataset is representative and classes, implicitly presented by the dataset,

I.A. Borisova (✉) • N.G. Zagoruiko
Sobolev Institute of Mathematics SD RAS, 4 Acad. Koptyug Avenue,
630090 Novosibirsk, Russian Federation
e-mail: biamia@mail.ru; zag@math.nsc.ru

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_9,

are compact. Under these conditions it is required to find a partition of the dataset V_u in K compact clusters. The clustering problem has a long history of researching. A considerable contribution in this domain was made by Boris Mirkin with his works [2, 3].

In the classification problem, to develop a decision rule, a set of training objects $V_l =< a_1, a_2, \ldots a_{Ml} >$ with given labels (class names) is used. It is supplemented with hypotheses that the training dataset is representative and classes are distributed compactly. Under these hypotheses clusters should be compact, which implies the compactness of both training objects and unseen unlabeled objects within the same class. Under these conditions it is required to find a classifier which recognizes both objects in the training dataset V_l and unlabeled objects with minimum errors. There are many specific methods for solving classification and clustering problems separately.

The problem of semi-supervised learning is intermediate between clustering and classification problems. It makes use of a mixture V_{mix} of labeled V_l and unlabeled V_u datasets for the analysis. It is supposed, that the mix dataset is representative and simultaneous distribution of labeled and unlabeled objects from the same classes is compact. Under these conditions it is required to find decision rule which divides the dataset V_{mix} into compact clusters and recognizes all the objects of the dataset—labeled and unlabeled—with minimum errors. When the volume of the labeled part V_l is insignificant or it is nonpresentative by itself, using of additional information from the unlabeled part V_u can considerably extend our idea of the general distribution and allows constructing decision rules with higher efficiency.

One of the approaches for solving the problem of semi-supervised learning is the co-training procedure [4], where two or more decision rules are generated according to the same set of objects, but in different independent feature spaces. An alternative approach is to model the joint probability distribution of the features and the labels. For the unlabeled data the labels can then be treated as "missing data." It is common to use the EM algorithm [5] to maximize the likelihood of the model. One more method for solving the problem of semi-supervised learning is implemented in the taxonomy decision rule (TDR) algorithm [1]. Its main idea is in using taxonomy algorithms for clustering the mixture of labeled and unlabeled objects into clusters, which should meet the conditions of geometrical compactness (the objects of one cluster are close to each other and far from the objects of other clusters) and uniformity (in one cluster the objects of one class are predominant). All three problems differ from each over in an input (only labeled objects, only unlabeled objects, mixture of labeled and unlabeled objects) but they are based on the same basic hypotheses that the dataset is representative and classes (given explicitly in supervised and semi-supervised learning or hidden as in clustering) are distributed compactly. This generality makes it possible to integrate all three problems into one problem of generalized classification, whose input dataset can include both labeled and unlabeled objects. If the number of the unlabeled objects is equal to zero this problem is reduced to the classification problem. If the number of the labeled objects is equal to zero the task is reduced to the clustering problem.

In this work, the FRiS-TDR algorithm is presented to tackle the problem of generalized classification solving. This algorithm is based on the function of rival similarity (FRiS-function) which efficiency for the classification and clustering problems has been demonstrated in our preview works [6,7]. In this paper definition of FRiS-function is extended in case of mixed datasets. Classifier, constructed with FRiS-TDR algorithm, consists of set of representative objects ("stolps"), placed in the centers of linearly separable clusters. Each object of the dataset is classified as a member of cluster rival similarity with which stolp is maximum. The clusters should be geometrically compact and homogeneous.

Compactness of classes and clusters, presented by training dataset, is a main requirement for the generalized classification problem. The block for compactness estimation is an important part of the FRiS-TDR algorithm. In [6] different ways to calculate compactness of clusters basing on rival similarity of the objects in the clusters were described. For the generalized classification problem the most appropriate definition of compactness appears to be the one, where members of each cluster should be similar to the stolp of that cluster.

At first we describe how to calculate FRiS-function, FRiS-compactness, and select sets of stolps for labeled datasets in the classification problem, for unlabeled dataset in the clustering problem, and then how to synthesize these ideas for semi-supervised learning and generalized classification problem.

2 Function of Rival Similarity

In many Data Mining problems the concept of "similarity," "closeness" is widely used. Many decision rules for the recognition of a new object are based on some measure of "similarity" of an object to a class. In the clustering problem, objects are united in the clusters of "similar" objects. More information about the importance of similarity in Data Miming one finds in [2,3].

But formal measures of "similarity" cannot be defined correctly without consideration of a context. So, Moscow and Washington will appear "close to each other," if one compares the distance between them with the distance from Moscow to the Sun, but "far from each over" if two cities within one state are considered as a standard of closeness.

So to estimate similarity of an object z to an object a correctly, we should take into account third object b, which determines the rival situation. If some distance measure d between objects is given, then a quantitative measure, which estimates rival similarity of z with a in competition with b, can be defined as follows:

$$F_{a/b}(z) = (d(z,b) - d(z,a))/(d(z,b) + d(z,a))$$

We had called this measure Function of Rival Similarity or FRiS-function [6]. FRiS-function takes values in range between -1 and $+1$. If the object z coincides with the object a, then similarity of these objects equals to 1. And if z coincides with

the object b, then similarity of z with a equals to -1. When value $F_{a/b}(z)$ equals to 0, i.e. $d(z,a) = d(z,b)$, it means the object z is equally similar (and is not similar) to both objects. The determined function of rival similarity is in good agreement with the similarity and difference perception mechanism intrinsic to human beings.

2.1 FRiS-function on the Labeled Dataset

In the classification problem, it is more important to estimate similarity of an object to a class, than with another object. If some labeled dataset V_1 consists of objects of two classes A and B, V^A—the set of objects of the class A, V^B—the set of objects of the class B, $V_1 = V^A \cup V^B$, then according to given assumptions to evaluate rival similarity of an object z with the class A it is necessary to consider not only distance r_A from z to this class, but also distance r_B to the nearest rival class (in case of two classes—distance from z to the class B). The measure of rival similarity of the object z to the class A in competition with the class B is defined as follows:

$$F_{a/b}(z) = (r_B(z) - r_A(z))/(r_B(z) + r_A(z))$$

As a distance from an object z to a class in the simplest case the distance from z to the nearest object of this class can be used, so $r_A(z) = \min\limits_{x \in V^A}\{d(z,x)\}$, $r_B(z) = \min\limits_{x \in V^B}\{d(z,x)\}$.

The next step of our research is in understanding how one can estimate compactness of classes with the help of FRiS-function. In case of classification problem, under FRiS-compactness of a class we assume average rival similarity of objects with the class. So to estimate FRiS-compactness of the labeled dataset V_1 at first for each object z of the dataset, we calculate the distance r_1 to its "own" class and the distance r_2 to the nearest "competing" class, i.e. for $z \in V^A$, $r_1 = r_A(z), r_2 = r_B(z)$, for $z \in V^B$ $r_1 = r_B(z), r_2 = r_A(z)$. The rival similarity of the labeled object z with its own class is:

$$F(z) = (r_2 - r_1)/(r_2 + r_1) \tag{1}$$

And then the value of FRiS-compactness of the labeled dataset V_1 is calculated as follows:

$$\overline{F}(V_1, V^A, V^B) = 1/|V_1| \sum_{z \in V_1} F(z)$$

This value can be used for estimating local compactness of classes. But our experience proves what using only typical representatives (stolps) of a dataset instead all objects gives better compactness estimation. In this case the distance from an object z to the nearest stolp of a class is used as the distance from z to

this class. If S^A is a set of stolps of the class A, and S^B is a set of stolps of the class B, then $r_A^S(z) = \min\limits_{s \in S^A}\{d(z,s)\}$, $r_B^S(z) = \min\limits_{s \in S^B}\{d(z,s)\}$, and for $z \in V^A$ $r_1^S = r_A^S(z)$, $r_2^S = r_B^S(z)$, for $z \in V^B$ $r_1^S = r_B^S(z)$, $r_2^S = r_A^S(z)$. With the formula (1) it is possible to calculate values of rival similarity:

$$F(z, S^A, S^B) = (r_2^S - r_1^S)/(r_2^S + r_1^S) \tag{2}$$

of all objects of sample V_1 with the set of stolps $S = S^A \cup S^B$. Averaging these values we receive the integrated characteristic:

$$\overline{F}(V_1, S^A, S^B) = 1/|V_1| \sum_{z \in V_1} F^S(z, S^A, S^B) \tag{3}$$

which estimates compactness of the dataset depending on the system of stolps S.

From another point of view this value determines how full S describes the whole dataset. The higher this value, the higher accuracy of the description is; the more typical objects are used as stolps. If from all possible systems of stolps of the dataset we somehow found an optimal system with the given volume L:

$$S^* = \arg\max_{|S|=L} \overline{F}(V_1, S),$$

then value $\overline{F}(V_1, S^*)$ can be used as a best estimation of compactness of the classes presented by the dataset V_1. And the system of stolps S^* in case of appropriate choice of L can be considered as an effective decision rule, suitable for recognition of new objects. Usually for this purpose L, equals to the minimal number of stolps, which is sufficient for correct recognition of all training objects, is used.

2.2 *Algorithm FRiS-Stolp for a Set of Stolps Forming on the Labeled Dataset*

To select the optimal or close to the optimal system of stolps S^* for description of the labeled dataset, algorithm FRiS-Stolp is used. Each stolp in the system has two peculiarities—*defensive capability* (high similarity with other objects from the same class, which allows recognizing these objects correctly) and *tolerance* (low similarity with the objects of the rival class, which prevents their misrecognizing as "own"). Basing on the set of stolps an unseen object z is classified as a member of a class, and similarity with that stolp is maximal. Value of rival similarity can be used as an estimation of reliability of the object z recognition.

This procedure of stolps selection is realized as follows:

1. Some object a_i of the class A is tested as a single stolp of this class. All objects of the class B are considered as the stolps of the rival class. As in compactness estimation values of similarity $F^S(a_j, \{a_i\} \cup V^B)$ of each object $a_j (j = 1, 2, \ldots, M_A)$ of the class A, and distinctiveness $F^S(b_q, V^B/\{b_q\}) \cup \{a_i\}$ of each object b_q, $(q = 1, 2, \ldots, M_B)$ of the class B with a_i are calculated according to formula (2) and added to the counter $C(a_i)$. Averaging value $C(a_i)$ is used as efficiency of the object a_i in a role of a stolp of the class A.
2. Step 1 is repeated for all objects of the class A. An object a which provides maximum value $C(a)$ is selected as a first stolp of the class A. All m_1 objects of the class, which similarity with this stolp is higher than F^* (for example, $F^* = 0$), form first cluster of the class A and are eliminated from the class A.
3. If $m_1 < M_A$ steps 1–2 are repeated on remaining objects of the class A. As a result a set S^A of k_A stolps of the class A is obtained. The dataset V^A is repartitioned into clusters by the following rule. Each object a joins the cluster, distance to which stolp is minimal.
4. Steps 1–3 are repeated for the class B to construct a list S^B of k_B stolps of this class. When the set of stolps S^* has been found we can estimate its compactness with formula (3).

In case of Gaussian distributions most typical objects of the classes, selected by algorithm FRiS-Stolp, are placed at the points of statistical means. In case of multimodal distributions and linearly inseparable classes stolps are placed at the centers of the modes (at the centers of areas of local concentrations of objects). With distributions complexity in growing the number of stolps increases.

2.3 FRiS-function on the Unlabeled Dataset

Peculiarity of FRiS-function calculation on the unlabeled dataset is in the fact that class names of training objects are unknown. All objects, as though, should be treated as belonging to a single class. If some system of stolps S for the dataset V_u description is given we can determine the distance r_1 from an object z to the "own" class as the distance from z to the nearest stolp from S:

$$r_1^S = \min_{s \in S} \{\rho(z, s)\}$$

But the absence of rival classes does not allow determining distance r_2 from the object z to the nearest stolp-competitor. In this case a virtual class-competitor is entered into consideration. Objects of this virtual class are settled on the fixed distance equal r^* from each object of V_u. As a result, instead of usual FRiS-function we use its modification, which for any object $z \in V_u$ is:

$$rF(z, S) = (r^* - r_1^S)/(r^* + r_1^S) \tag{4}$$

This modification is designated as the reduced function of rival similarity [7]. Average value of reduced function of rival similarity over all objects of the dataset V_{u}:

$$r\overline{F}(V_u, S) = 1/|V_u| \sum_{z \in V_u} rF(z, S) \tag{5}$$

characterizes how precisely and fully the system of stolps S describes the unlabeled dataset. If from all possible systems of stolps we somehow found an optimal system with maximal value of average reduced FRiS-function, that system could be used as a decision of clustering problem. The dataset V_{u} is partitioned into clusters by the following rule. Each object a joins that cluster, distance to which stolp is minimal.

2.4 Algorithm FRiS-Clust for a Set of Stolps Selection from the Unlabeled Dataset

To select the optimal or close to the optimal system of stolps S^* in case of an unlabeled datasets, algorithm FRiS-Clust is used. The described algorithm defines the number of clusters automatically. A user sets only the maximal number of clusters $k_{\max}$. The algorithm searches for decisions of a task for all numbers of clusters $k = 1, 2, \ldots, k_{\max}$ consistently, to choose from them the most successful decision. It works as follows:

1. Some object a_i of the dataset is tested as a single stolp of this dataset. As in compactness estimation values of similarity $rF(a_j, \{a_i\})$ of each object $a_j (j = 1, 2, \ldots, M)$ of the dataset V_{u} with this stolp in competition with a virtual competitor are calculated according to formula (4). Averaging value of similarity $r\overline{F}(V_{\text{u}}, \{a_i\})$ calculated according to (5) is used as efficiency of the object a_i in a role of stolp.
2. Step 1 is repeated for all objects of the dataset V_{u}. An object a which provide maximum value $r\overline{F}(V_{\text{u}}, \{a\})$ is selected as a first stolp s_1, $S_1 = \{s_1\}$.
3. Each object a_i of the dataset $(a_i \neq s_1)$ is tested as a second stolp of this dataset. Values $r\overline{F}(V_u, \{a_i, s_1\})$ of average similarity with this system of stolps in competition with a virtual competitor are calculated according to formula (5). An object a, which provide the maximum value is selected as a second stolp s_2, $S_2 = \{s_1, s_2\}$.
4. Number of stolps increases one by one in the same way until runs up to maximal number $k_{\max}$. As a result of each step the preliminary set of stolps S_k has been found. For each number of stolps k we can partition the dataset V_{u} into corresponded number of clusters.
5. When we selected stolp s_i of the system $S_k (k = 2, .., k_{\max})$, we did not take into account the information about all the stolps that would be selected after it. So for some clusters, we can find better positions of their stolps by the next procedure:

5.1. Each object a from the first cluster is tested to be a new stolp of the cluster. The best one is selected according to the criterion $\overline{F}(V_u, \{a, s_2, s_3, \ldots, s_k)$ calculated by formula (5). New position s_1^* of the stolp of the first cluster is fixed and compositions of the clusters are recalculated.

5.2. The best object to be a new stolp of the second cluster is selected according to criterion $\overline{F}(V_{\text{u}}, \{s_1^*, a, s_2, s_3, \ldots, s_k)$, new stolp is selected, and compositions of the clusters are recalculated as well. This procedure is repeated for each cluster until new position of each stolp from the system S_k is found.

5.3. As a quality of clustering into k clusters value $F_k = \overline{F}(V_{\text{u}}, \{s_1^*, s_2^*, \ldots, s_k^*)$ is used.

After the termination of the algorithm one variant of clustering for each number of clusters k with the calculated clustering quality F_k is found. Our experiments have shown that best variants of clustering are locally maximal, i.e. $(F_{k-1} < F_k)$ $\&(F_{k+1} < F_k)$. These variants experts are regarded as "reasonable" ones. "Reasonable" in this situation means that the objects, related to different clusters by the expert, are in different clusters formed by our algorithm.

3 Calculation of FRiS-function over Mixed Dataset

Consider now how the technique of rival similarity evaluation is changed in case of operating with mixed dataset V_{mix} consisting of an labeled V_{l} as unlabeled V_{u} objects. An unlabeled object is an object, for which class name is unknown and should be restored. In case of two classes A and B, objects of such mixed dataset V_{mix} can be divided into three sets $V_{\text{mix}} = V^A \cup V^B \cup V^C$. There V^A is a set of objects of the class A, V^B is a set of objects of the class B, and V^C—is a set of objects for which class name is unknown, i.e. unlabeled sample ($V^C = V_{\text{u}}$).

In the simplest case the distance from an object z to a class is calculated as the distance from z to the nearest object of this class. Presence of unlabeled objects V^C complicates calculating of these distances because for any object the nearest "own" and the nearest "rival" objects can be among a set of unlabeled objects. Therefore distance to the nearest "own" neighbour for $z \in V^A$ is $r_1 = \min\limits_{x \in V^A \cup V^C} \{d(z, x)\}$, for $z \in V^B$ is $r_1 = \min\limits_{x \in V^B \cup V^C} \{d(z, x)\}$.

For objects of sample V^C we consider that the nearest "own" for any object $z \in V^C$ can belong to any class, so $r_1 = \min\limits_{x \in V^A \cup V^B \cup V^C} \{d(z, x)\}$. Such approach is agreed with a hypothesis of the local compactness approving that close objects most likely belong to the same class.

To find the nearest competitor for taking into account the fact that it can be among unlabeled objects, we use the same technique, as in Sect. 2.3, add virtual competitor in consideration, and assign distance to it equal to r^*. In this case for object $z \in V^A$ $r_2 = \min\{r^*, \min\limits_{x \in V^B} \{d(z, x)\}\}$, for object $z \in V^B$ $r_2 = \min\{r^*, \min\limits_{x \in V^A} \{d(z, x)\}\}$.

To find distance r_2 from $z \in V^C$ to the nearest competitor we make a choice between distance r_A from z to the nearest object of the class A, distance r_B from z to the nearest object of the class B, and distance r^* to the virtual competitor. The minimal distance from first two values is supposed to be distance from z to the "own" class. Hence, distance to the competitor is equal to maximal of these two distances (r_A and r_B), or distance r^* to a virtual object. As a result for each object $z \in V^C$ $r_2 = \min\{r^*, \max\{r_A, r_B\}\}$.

Substituting the received values r_1 and r_2 in the formula for evaluation of rival similarity (1) we can calculate it for each object separately, and then receive the integrated characteristic for estimating compactness of the classes of the given mixed dataset.

Finally consider a case when the distance from an object z to a class is defined as the distance from z to the nearest stolp of this class. The system of stolps S for dataset V_{mix} in this case can be divided into two parts: S^A—the set of the stolps of the class A, S^B—the set of the stolps of the class B. Unlabeled objects from set V^C can be used as stolps of the class A, as stolps of the class B. Therefore the distance from an object $z \in V^A$ to the nearest "own" and nearest competitor from the set of stolps S is found as follows:

For object $z \in V^A$: $r_1^S = \min\limits_{s \in S^A}\{d(z,x)\}$, $r_2^S = \min\{r^*, \min\limits_{x \in S^B}\{d(z,x)\}\}$,

For object $z \in V^B$: $r_1^S = \min\limits_{s \in S^B}\{d(z,x)\}$, $r_2^S = \min\{r^*, \min\limits_{x \in S^A}\{d(z,x)\}\}$,

For object $z \in V^B$: $r_1^S = \min\limits_{s \in S^A \cup S^B}\{d(z,x)\}$, $r_2^S = \min\{r^*, \max\{\min\limits_{x \in S^A}\{d(z,x)\}, \min\limits_{x \in S^B}\{d(z,x)\}\}\}$.

Calculating values of rival similarity $F^{\text{mix}}(z,S)$ for each $z \in V_{\text{mix}}$ with the formula (2) and averaging them over all objects of the mixed dataset, we receive value:

$$F^{\text{mix}}(S) = 1/|V_{\text{mix}}| \sum_{z \in V_1 \cup V_u} F^{\text{mix}}(z,S) \tag{6}$$

This value is the characteristic of quality of the description of V_{mix} by system of stolps S.

Notice that average value of function of rival similarity calculated on a mixed dataset can be treated as criterion for a system of stolps S selection: it increases with increasing of geometrical compactness of received partition of the dataset into clusters, and with increasing in uniformity of the objects in the cluster. It is explained by the fact what in cluster of a class function of rival similarity for objects from the competing classes is negative. The less number of such objects in cluster, the higher value of criterion $F_{\text{mix}}(S)$ is. Therefore for solving the problem of generalized classification we should find the approached decision for next optimization task:

$$\overline{F}^{\text{mix}}(S) \to \max_S$$

If $V_l = \varnothing$, the given problem is reduced to the clustering problem, and if $V_u = \varnothing$—to the classification problem. Algorithm FRiS-TDR described in following section solves all the range of the problems from clustering through semi-supervised learning to classification using unified approach.

4 Algorithm FRiS-TDR

At the first stage of algorithm the base set of the stolps consisting of the best candidate to be a stolp of the class A and the best candidate to be a stolp of the class B is found. To estimate efficiency of object in a role of a stolp average value of function of rival similarity is calculated on the mixed dataset. Thus each class is considered as described by a single stolp, and the position of a stolp of the rival pattern varies—each time the nearest object from the rival class is used as a stolp-competitor. This procedure is realized as follows:

1. Some object $a \in V^A$ is tested as single stolp of the class A. In this condition distance from any object z to the nearest "own" and the nearest "another's" stolps are found by following rules.
   If $z \in V^A$: $r_1 = d(z,a), r_2 = \min\{r^*, \min\limits_{x \in V^B}\{d(z,x)\}\}$,
   If $z \in V^B$: $r_1 = \min\limits_{x \in V^B \cup V^C}\{d(z,x)\}, r_2 = \min\{r^*, d(z,a)\}$
   For an object $z \in V^C$ we calculate distance r_A from z to the nearest object of the class A and distance r_B from z to the nearest object of the class B.
   If $r_A < r_B$ the object z is supposed to belong to the class A according to hypothesis of local compactness. So as a distance to the "own" class the distance from z to a is taken. And as a distance to the rival class value r_B if it does not exceed r^* is taken. In such way: $r_1 = d(z,a), r_2 = \min\{r^*, r_B\}$.
   If $r_A > r_B$ the object z does not belong to the class A and the nearest object from sets V^B and V^C is treated as an "own" stolp for it. As a distance to rival pattern the distance to the object a if it does not exceed r^* is taken. In this case $r_1 = \min\limits_{x \in V^B \cup V^C}\{d(z,x)\}, r_2 = \min\{r^*, d(z,a)\}$.
   Basing on these distances, we calculate function of rival similarity $F^{\text{mix}}(z,S)$ for all objects of the dataset with the object a. Values $F^{\text{mix}}(z,S)$ for an objects z of the class A characterize protective abilities of the object a, and for an objects z of the class B,—tolerance of the object a to the rival pattern. Therefore, averaging $F^{\text{mix}}(z,S)$ over all objects of the mixed dataset, we calculate value F_a^A of efficiency of object a in a role of a stolp of the pattern A.
2. Step 1 is repeated for all objects of the class A.
3. The object $b \in V^B$ is tested as single stolp of the class B and distances from an object z to the nearest "own" and the nearest "rival" stolps are found by following rules.
   If $z \in V^A$: $r_1 = \min\limits_{x \in V^A \cup V^C}\{d(z,x)\}, r_2 = \min\{r^*, d(z,b)\}$,

If $z \in V^B$: $r_1 = d(z,b)$, $r_2 = \min\{r^*, \min\limits_{x \in V^A}\{d(z,x)\}\}$.

For an object $z \in V^C$ we calculate distance r_A from z to the nearest object of the class A and distance r_B from z to the nearest object of the class B.

If $r_A < r_B$: $r_1 = \min\limits_{x \in V^A \cup V^C}\{d(z,x)\}$, $r_2 = \min\{r^*, d(z,b)\}$.

If $r_A > r_B$: $r_1 = d(z,b)$, $r_2 = \min\{r^*, r_A\}$

Basing on these distances, we calculate function of rival similarity $F^{\text{mix}}(z)$ for all objects of the mixed dataset and, averaging it, we receive value F_b^B of efficiency of an object b in a role of a stolp of the class B

4. Step 3 is repeated for all objects of the class B.
5. Since class name for any unlabeled object c is unknown, at first it is tested in a role of a single stolp of the class A and its efficiency F_c^A is calculated. For this purpose Step 1 is repeated for this object. Then the object c is tested in a role of a single stolp of the class B and during running Step 3 value F_c^B is calculated.
6. Step 5 is repeated for all objects of class V^C.
7. An object $s_1 \in V^A \cup V^C$ which provides maximum value F_s^A is selected as the first stolp of the class A. As the first stolp of the class B the object $s_2 \in V^B \cup V^C$ which provides maximum value F_s^B is selected.

   Thus, at the first stage of work of the algorithm the system of stolps which contains two stolps is formed. At the second stage this system is widened till the necessary size. For estimating quality of the system of stolps S_k we can use directly formula (6). The algorithm of extending the stolps system looks as follows:
8. Some object $a \in V^A$ is added to the current system of stolps S_i consisting of i stolps as an additional stolp of the class A. With the formula (6) quality of this system $F_a^A = F^{\text{mix}}(S_i \cup \{a\})$ is calculated. This procedure is repeated for all objects of the class A.
9. Some object $b \in V^B$ is added to the current system of stolps S_i as an additional stolp of the class B. With the formula (6) quality of the system $F_b^B = F^{\text{mix}}(S_i \cup b)$ is calculated. This procedure is repeated for all objects of the class B.
10. Some object $c \in V^C$ is added the current system of stolps S_i as a new stolp of the class A and quality of this system F_c^A is calculated. Then the same object is considered as object of the second class and for this case the quality of the system F_c^B is calculated too. This procedure is repeated for all objects of class V^C.
11. The object s_{i+1} of the mixed sample V^{mix} in which addition to system of stolps S_i as much as possible improves its quality, is selected as $(i+1)$-th stolp. In the other words $S_{i+1} = S_i \cup s_{i+1}$, where $s_{i+1} = \text{argmax}\{\overline{F}_z^A, \overline{F}_z^B\}$.
12. Process of the stolps system (steps 8–12) extending is repeated until achievement of one of conditions of stop.

The most widespread condition of stop for decision rule in the form of stolps construction is in achieving that number of stolps which allows recognizing labeled

objects with the fixed accuracy. Demand of correct recognition often can lead to retraining. Other variant is in fixing the maximal admissible number of stolps in the system. As a criterion for stop of the algorithm in FRiS-TDR the same technique is used that has been used in a task of taxonomy for the definition of optimum number of clusters. For this purpose, for any object z its nearest stolp is considered as "own" and following on affinity as "competitor." The average value of FRiS-function F_i calculated under these conditions is compared with to the same values calculated for smaller and greater sets of stolps S_{i-1} and S_{i+1}. Fulfilment of condition $F_{i-1} < F_i$ and $F_{i+1} < F_i$ is considered as the indicator of the fact that the number of stolps equal i corresponds to one of the most preferable partition set of objects into clusters. If after termination of the algorithm work only unlabeled objects are presented in some clusters and it is possible to consider that clusters as realizations of new unknown patterns. Offered algorithm is linear. Its laboriousness has the order of complexity equals to $k_{\max} \cdot M^2 + N^2$, where M is the number of objects in the mixed dataset, N—is dimension of the features space, and $k_{\max}$ is the maximal number of stolps.

5 Examples of the Algorithm FRiS-TDR Working

For efficiency of the algorithm FRiS-TDR presentation we show the results of its work on the elementary two-dimensional tasks with small number of objects which one can solve "approximately" and compare his decision with the decision offered by algorithm. In first three figures universality of algorithm FRiS-TDR which solves the problem of generalized classification for the labeled dataset (Fig. 1), for the mixed dataset (Fig. 2) and for the unlabeled dataset (Fig. 3) is shown.

Hereinafter the objects of the class A are designated in figures by circles, the objects of the class B—by squares, and the objects, which class names are

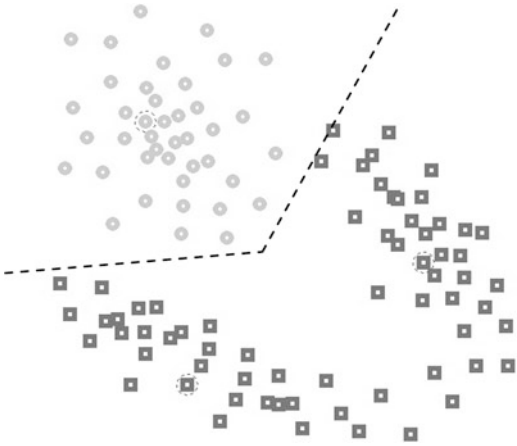


Fig. 1 Results of the application of the FRiS-TDR algorithm on the classification problem

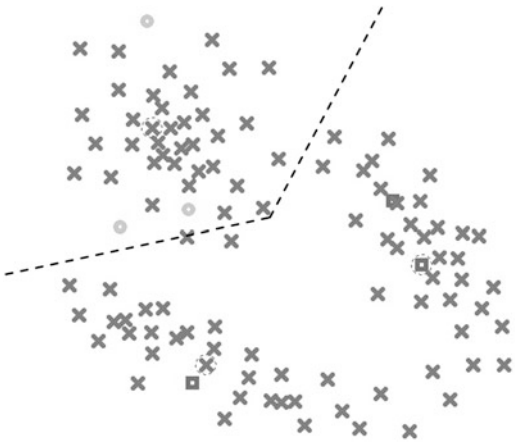


Fig. 2 Results of the application of the FRiS-TDR algorithm on generalized classification of the mixed dataset task

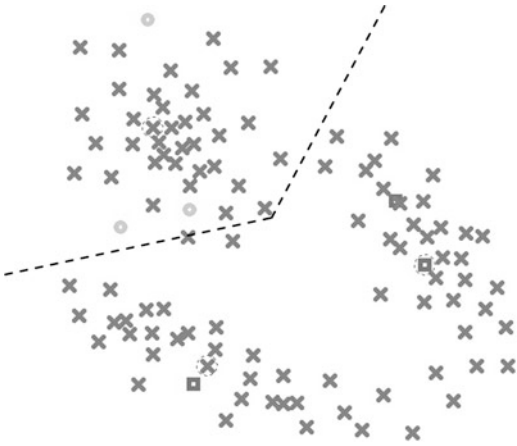


Fig. 3 Results of the application of the FRiS-TDR algorithm on the clustering problem

unknown,—by daggers. The objects selected by the algorithm as stolps are encircled by dotted lines, and borders for clusters are presented by dotted broken lines. Analyzing the presented results it is possible to see that given algorithm has found successful decisions for all three tasks.

For the illustration of the fact that construction of decision rule on the basis of the mixed dataset can improve quality of recognition, the task similar ones presented on Fig. 2 was solved, but unlabeled dataset was not used for decision rule construction. Results of this experiment are presented on Fig. 4. On it the border between classes constructed on mixed sample is presented by a dotted lines, and the border constructed only on labeled dataset—by continuous lines.

On Fig. 5 results obtained with the offered algorithm on the more complex task are presented. The number of stolps was determined automatically. Thus two clusters consisting of only unlabeled objects were allocated. These clusters can be considered as realizations of some unknown class

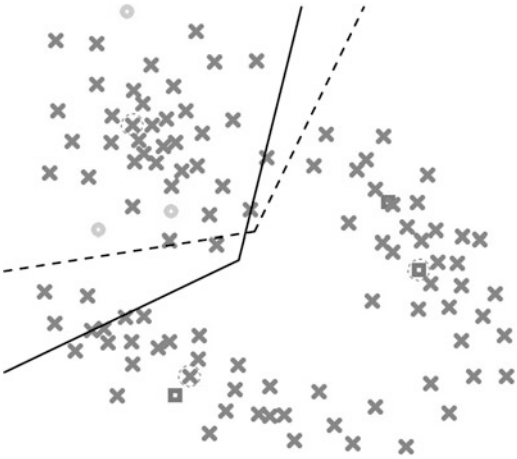


Fig. 4 The decision rules constructed on the mixed (a *dashed line*), and labeled (a *continuous line*) datasets

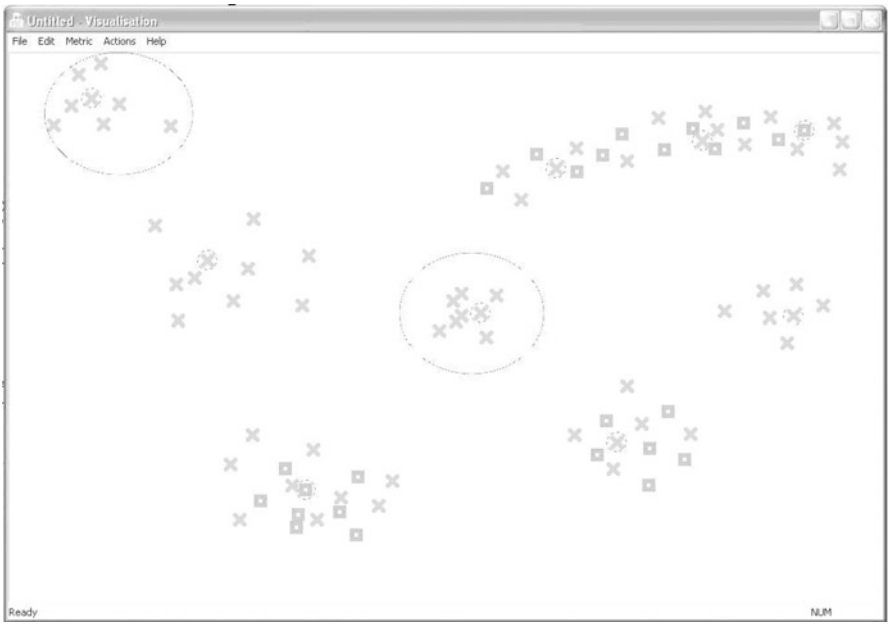


Fig. 5 Results of work of the algorithm FRiS-TDR on the complex mixed dataset

6 Conclusion

Here is a list of some results described in the paper:

1. A unified problem and algorithm for the generalized classification FRiS-TDR, that builds effective decision rules independently of proportion of labeled and unlabeled objects, are presented, in contrast to conventional approaches in which

tasks of analysis of labeled datasets, unlabeled datasets, and mixed datasets are considered different.
2. To select the best variant of the decision the value of FRiS-compactness of formed clusters is used.
3. Usage of FRiS-functions allows defining and finding the system of stolps, which represents the whole dataset fully and precisely.
4. It is shown that if only a small proportion of the dataset is labeled, then using the unlabeled part for building a decision rule enhances quality of the decisions.
5. Computational intensity of the algorithm FRiS-TDR does not exceed the intensities of FRiS-function based algorithms for solving clustering and classification problems separately.

Acknowledgements This study was conducted with partial financial support of the Russian Fund for Basic Research, the Project 11-01-00156.

References

1. Zagoruiko, N.G.: Applied Methods of the Data and Knowledge Analysis. Institute of Mathematics Press, Novosibirsk (1999)
2. Mirkin, B.: Core Concepts in Data Analysis: Summarization, Correlation, Visualization, 390 p. Springer, Berlin (2011)
3. Mirkin, B.: Clustering: A data Recovery Approach, 374 p. Chapman and Hall, London (2012)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Workshop on Computational Learning Theory, pp. 92–100. Morgan Kaufmann, Los Altos (1998)
5. Zhu, X., Goldberg, A.: Introduction to Semi-Supervised Learning. Morgan & Claypool, San Rafael (2009)
6. Borisova, I.A., Dyubanov, V.V., Zagoruiko, N.G., Kutnenko, O.A.: Use of FRiS-function for decision rule construction and attributes selection (a task of combined type DX). In: Proceedings of Conference on KONT-2007, Novosibirsk, vol. 1, pp. 37–44 (2007)
7. Borisova, I.A., Zagoruiko N.G.: Function of rival similarity in taxonomy task. In: Proceedings of Conference on KONT-2007, Novosibirsk, vol. 2, pp. 67–76 (2007)

From Separating to Proximal Plane Classifiers: A Review

Maria Brigida Ferraro and Mario Rosario Guarracino

Abstract A review of parallel and proximal plane classifiers is proposed. We discuss separating plane classifier introduced in support vector machines and we describe different proposals to obtain two proximal planes representing the two classes in the binary classification case. In details, we deal with proximal SVM classification by means of a generalized eigenvalues problem. Furthermore, some regularization techniques are analyzed in order to solve the singularity of the matrices. For the same purpose, proximal support vector machine using local information is handled. In addition, a brief description of twin support vector machines and nonparallel plane proximal classifier is reported.

Keywords Support vector machine • Proximal plane classifier • Regularized generalized eigenvalue classifier

1 Introduction

The aim of this paper is to describe and discuss different planes classifiers starting from a widely used learning algorithm: support vector machines (SVMs). The idea of SVMs was firstly introduced by Mangasarian [1] and later developed by Vapnik [2]. They are used as a learning technique, in particular in a classification framework. A classification algorithm is in essence an algorithm that learns

M.B. Ferraro (✉)
Department of Statistical Sciences, Sapienza University of Rome, High Performance Computing and Networking Institute, National Research Council, Naples, Italy
e-mail: mariabrigida.ferraro@uniroma1.it

M.R. Guarracino
High Performance Computing and Networking Institute,
National Research Council, Naples, Italy
e-mail: mario.guarracino@cnr.it

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_10,

(computes) a model from data divided in two or more classes. The obtained model is then used to assign the class label to a new unlabeled instance. In the case of two linearly separable classes, every classification task can be seen as a separation task, which reduces to the determination of a plane that leaves the points of the two classes in separate half spaces. The key point is that there can be infinitely many planes separating the classes, and so external conditions are needed to obtain a unique solution. The basic idea in SVMs is to choose two parallel planes separating the classes and maximizing the margin between them. The optimal separating plane is the midway plane. The solution is based on a quadratic programming problem (QPP) with linear constraints. SVM classifies unlabeled points by assigning them to the class in the corresponding half space. SVMs are used in many practical applications in economics and finance, biomedicine, bioinformatics, social sciences, and text categorization [3].

In case the classes are not linearly separable, the parallel planes are obtained maximizing a *soft* margin, that means some points are allowed to lay between the two parallel planes, by controlling their distance from the planes. Since the minimization problem is not twice differentiable, it is not possible to use a fast Newton method for the solution of the underlying QPP. Lee and Mangasarian [4] propose to consider an objective function with smoothing approximation. In this way, taking advantage of the twice differentiable property of an objective function with smooth terms, it is possible to use a quadratically convergent algorithm for solving the so called smooth support vector machine (SSVM).

In case of datasets belonging to large dimensional spaces, SVMs need to solve a quadratic program requiring an extensive computational time. As we will see in the following, lower computational complexity is traded against theoretical results regarding the generalization capability of the methods. Fung and Mangasarian [5] introduce a proximal support vector machine (PSVM) classifier obtained as a solution of a single system of linear equations. They propose to find two parallel planes, each one representing one class, such that the points of each class cluster around the plane, and they are as far as possible. In that case, the classification of unlabeled points is based on proximity to one of two parallel planes. PSVM leads to a small reduction of accuracy, that is still statistically comparable with that of standard SVM classifiers, and to a significantly lower computational time. Unfortunately, the theoretical error probability estimates for SVM are based on the margin and are not applicable anymore.

Since the objective is to find two planes representing two different classes, the idea of parallelism seems to be too restrictive and unrealistic. To this extent, Mangasarian and Wild [6] introduce a multisurface PSVM classification via generalized eigenvalues (GEPSVM). Each plane is obtained as the closest to one class and as far as possible from the other one. Dropping the parallelism condition, the classical binary XOR classification problem can be now solved. The optimization problem to obtain the planes is reduced to the minimization of a Rayleigh quotient, whose solution is obtained by solving a generalized eigenvalue problem, for which many well-known results exist in literature. Unfortunately, the matrices involved in the objective function can be singular, hence the solution can be not unique. There are different approaches to overcome this drawback. The first consists in introducing a

regularization term. Mangasarian and Wild [6] propose a Tikhonov regularization term in two generalized eigenvalue problems. Guarracino et al. [7] introduce a new regularization technique (ReGEC) in order to solve only one eigenvalues problem. A completely different approach is introduced by Yang et al. [8]. It solves the singularity problem arising in PSVM using local information (LIPSVM). In particular, LIPSVM is robust to outliers and the generation of the proximal planes is obtained by a standard eigenvalues problem. In addition, it is coherent from a geometric point of view, whereas the regularization terms make GEPSVM far from its original geometric interpretation.

In [9], another proximal planes classifier is introduced, known as twin support vector machine (TWSVM). Although the idea is similar to GEPSVM, the formulation is different. TWSVMs solve two QPPs, whereas SVMs solve only one QPP. In such sense is close to the idea of SVMs. A modification of TWSVM objective function is proposed by Ghorai et al. [10]. This proposal is called nonparallel plane proximal classifier (NPPC).

The rest of the paper is organized as follows. In next section the idea and the formulation of SVMs are recalled. Section 3 contains a description of Soft SVM, SSVM, and PSVM. In Sect. 4 GEPSVM is introduced and discussed. Furthermore, the problem of matrices singularity in GEPSVM is addressed. Section 5 contains a description of ReGEC and Sect. 6 deals with LIPSVM. TWSVM and NPPC are briefly recalled in Sect. 7. Finally, Sect. 8 contains some remarks and future directions.

The notation used in the paper is as follows. All vectors are indicated by lower case letters and matrices by capital letters. Any vector x is a column vector, unless transposed to row vector by a prime superscript x'. Given two vectors x and y in the n-dimensional real space $\mathbb{R}^n$, their scalar (inner) product is denoted by $x'y$, the 2-norm of x is $\|x\|$, a vector of ones of proper dimension is e.

2 Support Vector Machine

Let's consider a data set composed of k pairs (x_i, y_i), where $x_i \in \mathbb{R}^n$ is the feature vector that characterizes the point x_i and $y_i \in \{-1, 1\}$ is the class label. In a classification context, SVMs are used to find an hyperplane $\omega' x - \gamma = 0$, with orientation $\omega \in \mathbb{R}^n$ and relative location to the origin $\gamma \in \mathbb{R}$, with the aim to separate the elements belonging to two different classes. The idea consists in choosing two parallel planes, $x'\omega - \gamma = \pm 1$ which leave all points in separate half spaces and maximize the margin between the two classes. The margin μ can be defined as the distance between the planes:

$$\mu = \frac{2}{\|\omega\|}. \tag{1}$$

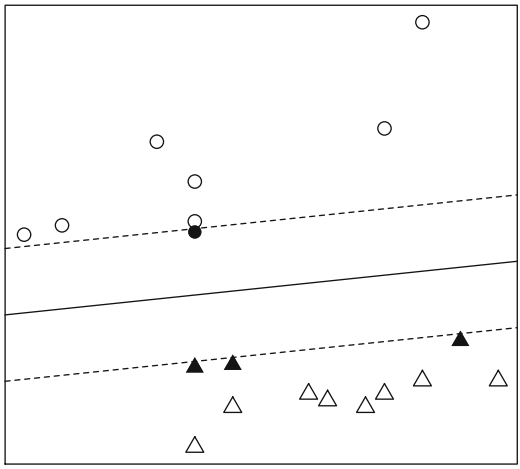


Fig. 1 Two classes perfectly linearly separable in a two-dimensional space: the standard support vector machine classifier (the *continuous line*) and the support vectors (the *bold* ones)

Then, the solution to the following quadratic linearly constrained problem is the optimal hyperplane with the maximum margin:

$$\min_{(\omega,\gamma)\in\mathbb{R}^{n+1}} \frac{\|\omega\|^2}{2}, \tag{2}$$

$$\begin{aligned} \text{s.t.} \quad & x_i'\omega - \gamma \geq 1, \quad y_i \in \text{class } 1, \\ & x_i'\omega - \gamma \leq -1, \; y_i \in \text{class } -1. \end{aligned} \tag{3}$$

The constraints (3) can be simplified to a single expression:

$$y_i(x_i'\omega - \gamma) \geq 1. \tag{4}$$

Only few points of the training set are needed to determine the hyperplane and they are called support vectors.

Figure 1 shows the hyperplane that separates the points of the two classes and the support vectors.

Considering two matrices $A \in \mathbb{R}^{p\times n}$ and $B \in \mathbb{R}^{m\times n}$, containing one feature vector on each row, that represent class 1 and class -1, respectively. The quadratic linearly constrained problem can also be written as:

$$\min_{(\omega,\gamma)\in\mathbb{R}^{n+1}} \frac{\|\omega\|^2}{2}, \tag{5}$$

$$\begin{aligned} \text{s.t.} \quad & (A\omega - e\gamma) \geq e, \\ & (B\omega - e\gamma) \leq -e. \end{aligned} \tag{6}$$

When the two classes are strictly linearly separable, the plane $x'\omega = \gamma + 1$ bounds all of the class 1 points, while the plane $x'\omega = \gamma - 1$ bounds all of the class -1 points as follows:

$$\begin{aligned} A\omega &\geq e\gamma + e, \\ B\omega &\leq e\gamma - e. \end{aligned} \tag{7}$$

Consequently, the plane:

$$x'\omega = \gamma, \tag{8}$$

midway between the bounding planes (7), is a separating plane that separates class 1 from class -1 completely.

Since the QPP is convex, any local minimum is global. The solution of this QPP can be obtained using de facto standard strategies such as those described in Morè and Toraldo [11].

3 Soft, Smooth, and Proximal Support Vector Machine

Standard SVMs classify points by assigning them to one of two disjoint half spaces. When the k points of two classes are not strictly linearly separable (Fig. 2) in the n-dimensional real space $\mathbb{R}^n$, the QPP (5) has no feasible solution. In such a case, it is possible to relax the constraints, allowing some points to fall the margin. A nonnegative slack variable $\xi = (\xi_1, \xi_{-1}) \in \mathbb{R}^{p+m}$ ($dim(\xi_1) = p$ and $dim(\xi_{-1}) = m$) is added to constraints and a penalty term to the objective function:

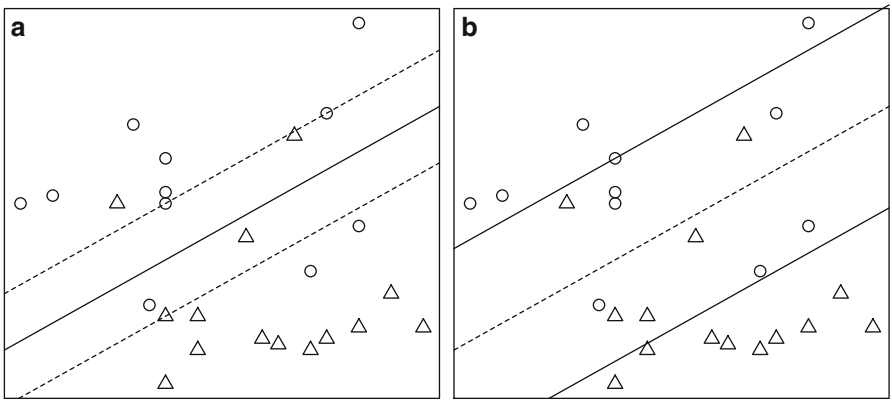


Fig. 2 Two classes not perfectly linearly separable in a two-dimensional space: (**a**) the separation obtained by soft SVM and (**b**) the separation obtained by PSVM

$$\min_{(\omega,\gamma,\xi)\in\mathbb{R}^{n+1+k}} ce'\xi + \frac{\|\omega\|^2}{2}, \tag{9}$$

$$\begin{aligned} \text{s.t.} \quad & (A\omega - e\gamma) + \xi_1 \geq e, \\ & (B\omega - e\gamma) - \xi_{-1} \leq -e, \\ & \xi \geq 0, \end{aligned} \tag{10}$$

If the classes are linearly separable, then $\xi = \underline{0}$. On the other hand, when they are linearly inseparable, which is the case shown in Fig. 2, then the two planes bound the two classes with a "soft margin" (i.e., bound approximately with some errors) determined by the nonnegative error variable ξ, that is:

$$\begin{aligned} A\omega + \xi_1 &\geq e\gamma + e, \\ B\omega - \xi_{-1} &\leq e\gamma - e. \end{aligned} \tag{11}$$

The 1-norm of the error variable ξ is minimized parametrically with weight c in (9) resulting in an approximate separating plane (8) as depicted in Fig. 2a (continuous line). This plane acts as a linear classifier as follows:

$$\begin{aligned} x'\omega - \gamma + \xi \geq 0 \quad & \text{then } x \text{ in class } 1, \\ x'\omega - \gamma - \xi \leq 0 \quad & \text{then } x \text{ in class } -1. \end{aligned} \tag{12}$$

In the smooth approach the square of 2-norm of the slack variable ξ is minimized with weight $c/2$, whereas the margin between the bounding planes is maximized with respect to both ω and γ, that is the distance between planes is, $\frac{1}{2}\| \begin{bmatrix} \omega \\ \gamma \end{bmatrix} \|^2$

$$\min_{(\omega,\gamma,\xi)\in\mathbb{R}^{n+1+k}} c\frac{1}{2}\|\xi\|^2 + \frac{1}{2}\| \begin{bmatrix} \omega \\ \gamma \end{bmatrix} \|^2, \tag{13}$$

$$\begin{aligned} s.t. \quad & (A\omega - e\gamma) + \xi_1 \geq e, \\ & (B\omega - e\gamma) - \xi_{-1} \leq -e, \\ & \xi \geq 0. \end{aligned} \tag{14}$$

It has been proven that the formulation (13) has the same performance of the classical SVM [4]. The problem (13) can be converted in an equivalent unconstrained problem:

$$\min_{(\omega,\gamma)\in\mathbb{R}^{n+1}} c\frac{1}{2}\left(\|(e - (A\omega - e\gamma))\|^2 + \|(e - (B\omega - e\gamma))\|^2\right) + \frac{1}{2}\| \begin{bmatrix} \omega \\ \gamma \end{bmatrix} \|^2. \tag{15}$$

Since the objective function in (15) is not twice differentiable, the smoothing techniques are applied and x is replaced by a very accurate smooth approximation $p(x,\alpha) = x + \frac{1}{\alpha}\log(1+\varepsilon^{-\alpha x})$ $(\alpha > 0)$. In this way a SSVM is obtained:

$$\min_{(\omega,\gamma)\in\mathbb{R}^{n+1}} c\frac{1}{2}\left(\|p(e-(A\omega-e\gamma),\alpha)\|^2 + \|p(e-(B\omega-e\gamma),\alpha)\|^2\right) + \frac{1}{2}\|\begin{bmatrix}\omega\\ \gamma\end{bmatrix}\|^2. \tag{16}$$

The computational time required for solving the quadratic program related to SVMs can be considerably long. To that extent, Fung and Mangasarian [5] suggest to use a PSVM classifier obtained as a solution of a single system of linear equations. Instead of SVMs, the aim is to find two parallel planes, each one representing one class. In that case, each point is classified on the basis of the proximity to one of two parallel planes that are moved as far apart as possible. The rates of classification accuracy are statistically comparable with those obtained by SVM classifiers. Furthermore, a significant reduction of computational times is reached. The main idea of PSVM is to replace the inequality constraints in (14) by equalities as follows:

$$\text{s.t.} \quad \begin{array}{l}(A\omega - e\gamma) + \xi_1 = e,\\ (B\omega - e\gamma) - \xi_{-1} = e.\end{array} \tag{17}$$

There is an important change in the model. Geometrically, this formulation is depicted in Fig. 2b, which can be interpreted as follows. The planes $x'\omega - \gamma = \pm 1$ are not bounding planes anymore, but can be seen as "proximal" planes, around which the points of each class are clustered, and that are pushed as far apart as possible by the term $(\omega'\omega + \gamma^2)$ in the objective function, which is the reciprocal of the 2-norm distance squared between the two planes in the (ω,γ) space $\mathbb{R}^{n+1}$.

4 Generalized Eigenvalue Proximal Support Vector Machine

Since the idea is to find two hyperplanes representing two different classes, it is useful and more realistic to drop the parallelism condition on the proximal planes, as proposed by Mangasarian and Wild [6]. The new formulation consists in seeking the plane

$$x'\omega_1 - \gamma_1 = 0 \tag{18}$$

in $\mathbb{R}^n$ closest to the points of class 1 and furthest from the points in class -1 and the plane

$$x'\omega_{-1} - \gamma_{-1} = 0 \tag{19}$$

closest to the points in class -1 and furthest from the points in class 1. The first plane is obtained by solving the following optimization problem:

$$\min_{(\omega,\gamma)\neq 0} \frac{\|A\omega - e\gamma\|^2 / \| \begin{bmatrix} \omega \\ \gamma \end{bmatrix} \|^2}{\|B\omega - e\gamma\|^2 / \| \begin{bmatrix} \omega \\ \gamma \end{bmatrix} \|^2}. \tag{20}$$

The minimization problem (20) is a ratio between the sum of squares of 2-norm distances in the (ω, γ)-space of points in class 1 to the plane representing this class and the sum of squares of 2-norm distances in the (ω, γ)-space of points in class -1 to the same plane. By simplifying (20) we obtain:

$$\min_{(\omega,\gamma)\neq 0} \frac{\|A\omega - e\gamma\|^2}{\|B\omega - e\gamma\|^2}. \tag{21}$$

With the following positions:

$$\begin{aligned} G &:= [A \quad -e]'[A \quad -e], \\ H &:= [B \quad -e]'[B \quad -e], \\ z &:= \begin{bmatrix} \omega \\ \gamma \end{bmatrix}, \end{aligned} \tag{22}$$

the optimization problem (21) becomes:

$$\min_{z\neq 0} \frac{z'Gz}{z'Hz}, \tag{23}$$

where G and H are symmetric matrices in $\mathbb{R}^{(n+1)\times(n+1)}$. The objective function of (23) is known as the Rayleigh quotient of the generalized eigenvalue problem:

$$Gz = \lambda Hz. \tag{24}$$

The inverse of the objective function in (23) has the same eigenvectors and reciprocal eigenvalues. In [6] it is proven that proximal planes are defined by

$$z_{\min} = [\omega_1 \quad \gamma_1]', \quad z_{\max} = [\omega_{-1} \quad \gamma_{-1}]' \tag{25}$$

where $z_{\min}$ and $z_{\max}$ are the eigenvectors related to the eigenvalues of smallest and largest modulo, respectively. Then, $x'\omega_1 - \gamma_1 = 0$ is the closest hyperplane to the set of points in class 1 and the furthest from those in class -1, in the same way, $x'\omega_{-1} - \gamma_{-1} = 0$ is the closest hyperplane to the set of points in class -1 and the furthest from those in class 1.

Dropping the parallelism condition on the planes enables to solve problems that are not linearly separable, such as the XOR case depicted in Fig. 3.

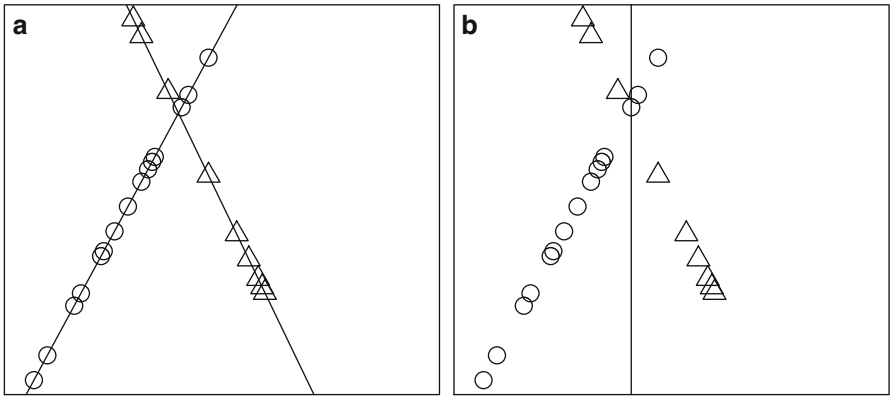


Fig. 3 XOR case: (**a**) GEPSVM solution and (**b**) PSVM solution

GEPSVM relaxes the parallelism and aims at obtaining two nonparallel planes from two corresponding generalized eigenvalue problems, respectively. However, it faces with the singularity problem. Since the matrices G and H can be deeply rank deficient, there is a non-zero probability that the null spaces of the two matrices have a non trivial intersection. In order to solve this problem there are two solutions:

1. to introduce a regularization technique to be applied in order to numerically solve the problem,
2. to consider a classifier based on local information and obtain as solution an ordinary eigen-system.

5 Regularized Generalized Eigenvalue Classifier

Mangasarian and Wild propose to regularize the problem, as is often done in least squares and mathematical programming problems [12, 13], by means of a Tikhonov regularization term [14]. It consists in reducing the norm of the variables (ω, γ) that determine the proximal planes (18) and (19). That is, for nonnegative parameter δ, problem (21) is regularized in the following way:

$$\min_{(\omega,\gamma)\neq 0} \frac{\|A\omega - e\gamma\|^2 + \delta\|z\|^2}{\|B\omega - e\gamma\|^2}, \tag{26}$$

and the proximal hyperplane related to the other class can be obtained as a solution of

$$\min_{(\omega,\gamma)\neq 0} \frac{\|B\omega - e\gamma\|^2 + \delta\|z\|^2}{\|A\omega - e\gamma\|^2}. \tag{27}$$

From a geometric point of view, the plane solution of Eq. (26) is the closest plane to the data set represented by A, normalized by the sum of the distances to the points of B. Simultaneously, the plane obtained by (27) is the closest one to the data set represented by B, normalized by the sum of the distances to the points of A. Guarracino et al. [7] give a more flexible technique for the regularization parameter in the kernel case and name so-proposed plane classifier as ReGEC (Regularized Generalized Eigenvalue Classifier). ReGEC simultaneously finds two planes from a single generalized eigenvalue equation (the two planes correspond, respectively, to the maximal and minimal eigenvalues), instead of two equations as in GEPSVM. In the linear case the new regularization method consists in solving the following generalized eigenvalue problem:

$$\min_{(\omega,\gamma)\neq 0} \frac{\|A\omega - e\gamma\|^2 + \delta\|\tilde{B}\omega - e\gamma\|^2}{\|B\omega - e\gamma\|^2 + \delta\|\tilde{A}\omega - e\gamma\|^2}. \tag{28}$$

Here $\tilde{A}$ and $\tilde{B}$ are diagonal matrices whose entries are the main diagonals of the A and B, respectively. This regularization provides classification accuracy results comparable to the ones obtained by solving Eqs. (26) and (27) and it is a form of robustification [15].

6 Proximal Support Vector Machine using Local Information

The classifiers introduced in the previous section adopt a regularization technique. The introduction of the regularization term in GEPSVM goes away from its original geometric interpretation. Yang et al. [8] propose a LIPSVM, whose solution is just an ordinary eigen-system. LIPSVM consists of two steps. In the first step, *interior* and *marginal* points are selected as belonging to the intra-class and the inter-class graphs. The intra-class graph is composed of edges connecting data points that are mutually k_1-nearest neighbors (k_1-NN) for a fixed k_1. The resulting graph is composed of a subset of points from the same class, that can be characterized as interior points. On the other hand, the inter-class graph is composed of edges connecting pairs of samples from different classes where one is a k_2-NN of the other. The interior points that lay in high-density regions become more likely nonzero-degree vertices, while the outliers that lay in low-density regions become zero-degree points. On the other hand, the marginal points are probably more nonzero-degree vertices.

In the second step, only those nonzero-degree points are used to train classifier. Thus, LIPSVM can dominate outliers (see Fig. 4). LIPSVM just requires solving a standard eigenvalue problem, whereas GEPSVM needs to solve a generalized eigenvalue problem. Furthermore, the number of the selected points used to train LIPSVM is smaller than that of the GEPSVM.

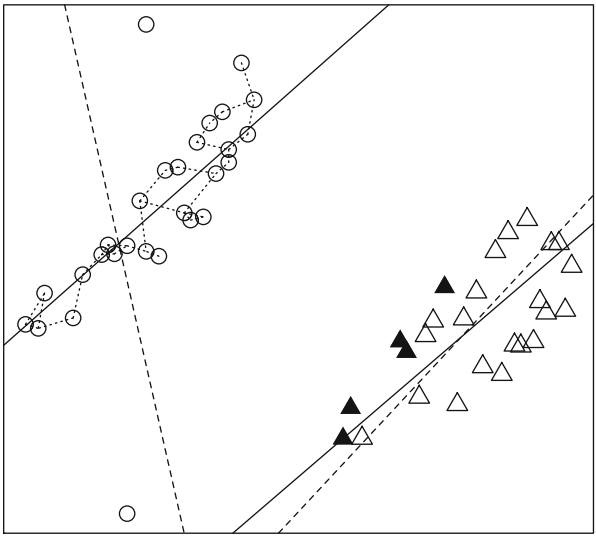


Fig. 4 The planes obtained by GEPSVM (*dotted lines*) and those obtained by LIPSVM (*continuous lines*). The intra-class graph for points represented by *circles* (class 1) and marginal points (*bold triangles*) corresponding to class −1

As in Fig. 4, the two adjacent matrices of each plane are, respectively, denoted by S and R and defined as follows:

$$S_{lt} = \begin{cases} \tau_l (> 0) & x_l \in Ne_{k_1}^{+}(t) \wedge x_t \in Ne_{k_1}^{+}(l), \\ 0 & \text{otherwise}, \end{cases} \tag{29}$$

and

$$R_{lt} = \begin{cases} \phi_t (> 0) & x_t \in Ne_{k_2}^{-}(l), \\ 0 & \text{otherwise}, \end{cases} \tag{30}$$

where $Ne_{k_1}^{+}(t)$ indicates a set of the k_1-NN in the same class of the sample x_l, and $Ne_{k_2}^{-}(l)$ a set of data points composed of k_2-NN in the different class of the sample x_l. When $S_{lt} > 0$ or $R_{lt} > 0$, an edge between x_l and x_t is inserted in the corresponding graph. By using non-zero degree vertices, a linear plane of LIPSVM can be constructed. The optimal plane of LIPSVM representing class 1 is obtained by solving the following minimization problem:

$$\min_{(\omega_1,\gamma_1)\in\mathbb{R}^{n+1}} \sum_{j=1}^{p} \sum_{l=1}^{p} S_{jl}(\omega_1' A_l - \gamma_1)^2 - \sum_{j=1}^{p} \sum_{t=1}^{m} R_{jt}(\omega_1' B_t - \gamma_1)^2, \tag{31}$$

$$\text{s.t.} \quad \|\omega_1\| = 1,$$

where A_l is the l-row of A and B_t the t-row of B. By using the weights $d_l = \sum_{j=1}^{m} S_{jl}$ and $f_t = \sum_{j=1}^{p} R_{jt}$ ($l = 1, \cdots, p$, and $t = 1, \cdots, m$), problem (31) can be simplified as follows:

$$\min_{(\omega_1,\gamma_1)\in\mathbb{R}^{n+1}} \sum_{l=1}^{p} d_l(\omega_1' A_l - \gamma_1)^2 - \sum_{t=1}^{m} f_t(\omega_1' B_t - \gamma_1)^2. \tag{32}$$

This optimization problem is reduced to a standard eigenvalue problem. Since the effect of outliers is eliminated or restrained LIPSVM is a robust method. In detail, A_l will be present in (32) if and only if its weight d_l is greater than 0 and, analogously, B_t, its corresponding marginal point, will be involved in (32) when $f_t > 0$. The points kept in the optimization problem are generated by S and R. In most cases the number of the marginal samples is lower than the number of the original ones. The expression $(\omega_1' x - \gamma_1)^2$ in (32) is the square distance of the points to the plane $x'\omega_1 - \gamma_1 = 0$. Thus, the aim of LIPSVM is to look for the plane $x'\omega_1 - \gamma_1 = 0$ closest to the interior samples in class 1 and furthest from the marginal samples in class -1.

7 Twin Support Vector Machine and Linear Nonparallel Plane Proximal Classifier

Jayadeva et al. [9] propose a novel approach to SVM classification, namely TWSVMs, which are similar to GEPSVMs in that they obtain nonparallel planes prototyping the data points. Indeed, they are based on a completely different formulation. Each of the two QPPs in the TWSVM pair has the formulation of a typical SVM, but not all samples appear in the constraints of each problem at the same time. The TWSVM classifier is obtained by solving the following pair of QPPs:

$$\begin{aligned} &\min_{(\omega_1,\gamma_1,\xi_{-1})\in\mathbb{R}^{n+1+m}} \tfrac{1}{2}\|A\omega_1 - e_1\gamma_1\|^2 + c_1 e_{-1}'\xi_{-1}, \\ &\text{s.t.} \quad -(B\omega_1 - e_{-1}\gamma_1) + \xi_{-1} \geq e_{-1}, \quad \xi_{-1} \geq 0, \end{aligned} \tag{33}$$

and

$$\begin{aligned} &\min_{(\omega_{-1},\gamma_{-1},\xi_1)\in\mathbb{R}^{n+1+p}} \tfrac{1}{2}\|B\omega_{-1} - e_{-1}\gamma_{-1}\|^2 + c_{-1} e_1'\xi_1, \\ &\text{s.t.} \quad (A\omega_{-1} - e_1\gamma_{-1}) + \xi_1 \geq e_1, \quad \xi_1 \geq 0, \end{aligned} \tag{34}$$

where c_1, $c_{-1} > 0$ are parameters. The algorithm finds two hyperplanes, one for each class, and classifies points on the basis of the distance of a given point to the hyperplane. The first term in the objective function of (33) or (34) is the

sum of squared distances from the hyperplane to points of one class. Therefore, its minimization tends to keep the hyperplane close to points of one class. The constraints lead the hyperplane to be at a distance of at least 1 from points of the other class. A set of slack variables is used to measure the error wherever the hyperplane is closer than this minimum distance of 1. The second term of the objective function minimizes the sum of slack variables, thus acting to minimize misclassification due to points belonging to the other class.

Ghorai et al. [10] propose a modification of the objective function of TWSVM. In details, they introduce a NPPC. It aims at putting together the idea of both TWSVM and PSVM. In the linear case (LNNPC) the formulation is the following

$$\min_{(\omega_1,\gamma_1,\xi_{-1})\in\mathbb{R}^{n+1+m}} \frac{1}{2}\|A\omega_1 - e_1\gamma_1\|^2 + c_1 e'_{-1}\xi_{-1} + \frac{c_2}{2}\|\xi_{-1}\|^2,$$
$$\text{s.t.} \quad -(B\omega_1 - e_{-1}\gamma_1) + \xi_{-1} \geq e_{-1}, \quad \xi_{-1} \geq 0, \tag{35}$$

and

$$\min_{(\omega_{-1},\gamma_{-1},\xi_1)\in\mathbb{R}^{n+1+p}} \frac{1}{2}\|B\omega_{-1} - e_{-1}\gamma_{-1}\|^2 + c_3 e'_1\xi_1 + \frac{c_4}{2}\|\xi_1\|^2,$$
$$\text{s.t.} \quad (A\omega_{-1} - e_1\gamma_{-1}) + \xi_1 \geq e_1, \quad \xi_1 \geq 0, \tag{36}$$

where c_1, c_2, c_3, and c_4 are the regularization parameters. Naturally, the introduction of those parameters gives rise to a problem related to the time needed for model selection, which can only be solved by prior knowledge about the application. On the other hand the latter methods seem to be suited to solve problems in which support vectors and inequality constraints can be neglected.

8 Concluding Remarks

In this paper we review planes classifiers starting from the idea of parallel planes of SVMs. Putting in evidence the limits of parallelism, we describe different proposals of classifiers that overcome this point. Even if in the last years the use of nonparallel plane classifiers has increased, there are still many open problems. In particular, in the drift from discriminating to proximal plane classifier, the concept of classification is substituted with that of characterization, and the objective is not anymore to minimize a classification error, but rather to describe a set of samples. Reintroducing the concept of minimum classification error would greatly improve accuracy performances of those methods. Furthermore, the introduction of different norms in the underlying mathematical programming problem is a crucial point. It would be interesting to determine the impact of a new norm on the results. Finally, as for SVMs, it would be of great interest to define and discuss an analytic formulation for the generalization error in the case of nonparallel plane classifiers.

Acknowledgements Authors would like to thank Dr. Panos Pardalos for the fruitful discussions and advice. This work has been partially funded by Italian Flagship project *Interomics* and by the Italian Ministry of Education, University and Research grant PON00619.

References

1. Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. Oper. Res. **13**, 444–452 (1965)
2. Vapnik, V.: Estimation of Dependences Based on Empirical Data [in Russian]. Nauka, Moscow (1979). (English translation: Springer, New York, 1982)
3. Pardalos, P.M., Hansen, P. (eds.): Data Mining and Mathematical Programming. CRM Proceedings and Lecture Notes, American Mathematical Society. vol. 45 (2008)
4. Lee, Y.-J., Mangasarian, O.L.: SSVM: A smooth support vector machine for classification. Comput. Optim. Appl. **20**, 5–22 (2001)
5. Fung, G., Mangasarian, O.L.: Proximal support vector machine classifiers. In: Proceedings KDD-2001: Knowledge Discovery and Data Mining (KDD 2001), San Francisco, CA (2001)
6. Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 69–74 (2006)
7. Guarracino, M.R., Cifarelli, C., Seref, O., Pardalos, P.: A classification method based on generalized eigenvalue problems. Optim. Methods Softw. **22**, 73–81 (2007)
8. Yang, X., Chen, S., Chen, B., Pan, Z.: Proximal support vector machine using local information. Neurocomputing **73**, 357–365 (2009)
9. Jayadeva, Khemchandani, R., Chandra, S.: Twin Support Vector Machines for Pattern Classification. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 905 (2007)
10. Ghorai, S., Mukherjee, A., Dutta, P.K.: Nonlinear plane proximal classifier. Signal Process. **89**, 510–522 (2009)
11. Morè, J.J., Toraldo, G.: On the solution of large quadratic programming problems with bound constraints. SIAM J. Opt. **1** (1991) 93–113.
12. Mangasarian, O.L., Meyer, R.R.: Nonlinear perturbation of linear programs. SIAM J. Control Optim. **17**, 745–752 (1979)
13. Mangasarian, O.L.: Least norm solution of non-monotone complementarity problems. In: Functional Analysis, Optimization and Mathematical Economics, pp. 217–221. New York, Oxford University Press (1990)
14. Tikhonov, A.N., Arsen, V.Y.: Solutions of Ill-Posed Problems. Wiley, New York (1977)
15. Xanthopoulos, P., Guarracino, M.R., Pardalos, P.M.: Robust generalized eigenvalue classifier with ellipsoidal uncertainty. Ann. Oper. Res. **216**, 327–342 (2014)

A Note on the Effectiveness of the Least Squares Consensus Clustering

Boris Mirkin and Andrey Shestakov

Abstract We develop a consensus clustering framework proposed three decades ago in Russia and experimentally demonstrate that our least squares consensus clustering algorithm consistently outperforms several recent consensus clustering methods.

Keywords Consensus clustering • Ensemble clustering • Least squares

1 Introduction

The problem of finding a partition reconciling a set of pre-specified partitions has been stated, developed and applied by Mirkin and Cherny in the beginning of the 1970s in the context of "nominal factor analysis" [2,3,8,9]. Yet this work remained largely unknown until Meila [7] mentioned the so-called Mirkin's distance, a tip of the iceberg of the work.

Perhaps the grand start for a consensus clustering approach on the international scene was made by Strehl and Ghosh [15]. Since then consensus clustering has become popular in bioinformatics, web-document clustering and categorical data analysis. According to [5], consensus clustering algorithms can be organized in three main categories: probabilistic approach [16,17]; direct approaches [1,4,14,15], and pairwise similarity-based approach [6, 11]. The (i,j)-th entry a_{ij} in the consensus matrix $A = (a_{ij})$ shows the number of partitions in which objects y_i and y_j are in the same cluster.

Here we invoke a least-squares consensus clustering approach from the paper [12] predating the above developments, update it with a more recent clustering

B. Mirkin (✉) • A. Shestakov
Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, 20 Myasnitskaya Ulitsa, Moscow 101000, Russian Federation
e-mail: bmirkin@hse.ru; ashestakoff@hse.ru

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_11,

procedure to obtain an algorithm for consensus clustering and compare the results on synthetic data of Gaussian clusters with those by the more recent methods. It appears our method outperforms those with a good margin.

2 Least Squares Criterion for Consensus Clustering

Given a partition of N-element dataset Y on K non-overlapping classes $S = \{S_1, \ldots, S_K\}$, its binary membership $N \times K$ matrix $Z = (z_{ik})$ is defined so that $z_{ik} = 1$ if y_i belongs to S_k and $z_{ik} = 0$, otherwise. As is known, the orthogonal projection matrix over the linear space spanning the columns of matrix Z is defined as $P_Z = Z(Z^T Z)^{-1} Z^T = (p_{ij})$ where $p_{ij} = \frac{1}{N_k}$, if $y_i, y_j \in S_k$ and 0 otherwise.

Given a profile of T partitions $R = \{R^1, R^2, \ldots, R^T\}$, its ensemble consensus partition is defined as that with a matrix Z minimizing the sum of squared residuals in equations

$$x_{il}^t = \sum_{k=1}^{K} c_{kl}^t z_{ik} + e_{ik}^t, \tag{1}$$

over the coefficients c_{kl}^t and matrix elements z_{ik} where $X^t, t = 1, \ldots, T$ are binary membership matrices for partitions in the given profile R. The criterion can be equivalently expressed as

$$E^2 = \|X - P_Z X\|^2, \tag{2}$$

where X is concatenation of matrices $X^1, \ldots, X_t$ and $\|\cdot\|^2$ denotes the sum of squares of the matrix elements. This can be further transformed into an equivalent criterion to be maximized:

$$g(S) = \sum_{k=1}^{K} \sum_{i,j \in S_k} \frac{a_{ij}}{N_k}, \tag{3}$$

where $A = (a_{ij})$ is the consensus matrix A from the pairwise similarity-based approach.

To (locally) maximize (3), we use algorithm AddRemAdd(j) from Mirkin in [10] which finds clusters one-by-one. Applied to each object y_j this method outputs a cluster with a high within cluster similarity according to matrix A. AddRemAdd(j) runs in a loop over all $j = 1 \ldots N$ and takes that of the found clusters at which (3) is maximum. When it results in cluster $S(j)$, the algorithm is applied on the remaining dataset $Y' = Y \backslash S(j)$ with a correspondingly reduced matrix A'. It halts when no unclustered entities remain. The least squares ensemble consensus partition consists of the AddRemAdd cluster outputs: $S^* = \bigcup S(j)$. It should be pointed out that the number of clusters is not pre-specified at AddRemAdd.

3 Experimental Results

All evaluations are done on synthetic datasets that have been generated using Netlab library [13]. Each of the datasets consists of 1,000 twelve-dimensional objects comprising nine randomly generated spherical Gaussian clusters. The variance of each cluster lies in 0.1–0.3 and its center components are independently generated from the Gaussian distribution $\mathcal{N}(0, 0.7)$.

Let us denote the thus generated partition as Λ with $k_\Lambda = 9$ clusters. The profile of partitions $R = \{R^1, R^2, \ldots, R^T\}$ for consensus algorithms is constructed as a result of $T = 50$ runs of k-means clustering algorithm starting from random k centers. We carry out the experiments in four settings: (a) $k = 9 = k_\Lambda$, (b) $k = 6 < k_\Lambda$, (c) $k = 12 > k_\Lambda$, (d) k is uniformly random on the interval $(6, 12)$. Each of the settings results in 50 k-means partitions. After applying consensus algorithms, adjusted rand index (ARI) [5] for the consensus partitions S and generated partition Λ is computed as $\varphi^{\text{ARI}}(S, \Lambda)$.

3.1 Comparing Consensus Algorithms

The least squares consensus results have been compared with the results of the following algorithms (see Tables 1, 2, 3, and 4):

- Voting Scheme (Dimitriadou, Weingessel and Hornik—2002) [4]
- cVote (Ayad—2010) [1]

Table 1 The average values at, $\phi^{\text{ARI}}(S, \Lambda)$ and the number of classes at $k_\Lambda = k = 9$ over 10 experiments in each of the settings

Algorithm	Average ϕ^{ARI}	Std. ϕ^{ARI}	Avr. # of classes	Std. # of classes
ARA	**0.9578**	0.0246	7.6	0.5164
Vote	0.7671	0.0624	8.9	0.3162
cVote	0.7219	0.0882	8.1	0.7379
Fus	0.7023	0.0892	11.6	1.8379
Borda	0.7938	0.1133	8.5	0.7071
MCLA	0.7180	0.0786	8.6	0.6992

Table 2 The average values of $\phi^{\text{ARI}}(S, \Lambda)$ and the number of classes at $k_\Lambda > k = 6$ over 10 experiments in each of the settings

Algorithm	Average ϕ^{ARI}	Std. ϕ^{ARI}	Avr. # of classes	Std.# of classes
ARA	**0.8333**	0.0586	6.2	0.6325
Vote	0.7769	0.0895	5.9	0.3162
cVote	0.7606	0.0774	5.6	0.6992
Fus	**0.8501**	0.1154	7.7	1.3375
Borda	0.7786	0.0916	6	0
MCLA	0.7902	0.0516	6	0

Table 3 The average values of $\phi^{\mathrm{ARI}}(S, \Lambda)$ and the number of classes at $k_\Lambda < k = 12$ over 10 experiments in each of the settings

Algorithm	Average ϕ^{ARI}	Std. ϕ^{ARI}	Avr. # of classes	Std.# of classes
ARA	**0.9729**	0.0313	9	0.9428
Vote	0.6958	0.0796	11.4	0.5164
cVote	0.672	0.0887	10.9	0.7379
Fus	0.6339	0.0827	16	4
Borda	0.7132	0.074	11.1	0.7379
MCLA	0.6396	0.0762	11.9	0.3162

Table 4 The average values of $\phi^{\mathrm{ARI}}(S, \Lambda)$ and the number of classes at $k \in (6, 12)$ over 10 experiments in each of the settings

Algorithm	Average ϕ^{ARI}	Std. ϕ^{ARI}	Avr. # of classes	Std.# of classes
ARA	**0.9648**	0.019	6.8	0.7888
cVote	0.5771	0.1695	10.4	1.2649
Fus	0.62	0.0922	11.6	2.0656
MCLA	0.6567	0.1661	10.6	1.3499

- Fusion Transfer (Guenoche—2011) [6]
- Borda Consensus (Sevillano, Carrie and Pujol—2008) [14]
- Meta-CLustering Algorithm (Strehl and Ghosh—2002) [15]

Tables 1, 2, 3, and 4 consistently show that:

- The least-squares consensus clustering algorithm has outperformed the other consensus clustering algorithms consistently;
- The only exception, at option (c), with $k_\Lambda > k = 6$ the Fusion Transfer algorithm demonstrated a better result probably because of the transfer procedure (see Table 2).
- The average number of clusters in the consensus clustering is lower than k in the profile R and k_Λ

4 Conclusion

This paper revitalizes a 30-years-old approach to consensus clustering proposed by Mirkin and Muchnik in Russian. When supplemented with updated algorithmic procedures, the method shows a very good competitiveness over a set of recent cluster consensus techniques. Our further work will include: (a) extension of the experimental series to a wider set of consensus clustering procedures, including those based on probabilistic modeling, (b) attempts at using the approach as a device for choosing "the right number of clusters," (c) exploring various devices, such as random initializations in k-means or bootstrapping of variables, for generation of ensembles of partitions, etc.

Acknowledgements This work was supported by the research grant "Methods for the analysis and visualization of texts" No. 13-05-0047 under The National Research University Higher School of Economics Academic Fund Program in 2013.

References

1. Ayad, H., Kamel, M.: On voting-based consensus of cluster ensembles. Pattern Recognit. **43**(5), 1943–1953 (2010)
2. Cherny, L.B.: The method of the partition space in the analysis of categorical features. A Ph.D. thesis, Institute of Control Problems, Moscow (1973) (in Russian)
3. Cherny, L.B.: Relationship between the method of the partition space and other methods of data analysis. In: Mirkin, B. (ed.) Issues in Analysis of Complex Systems, pp. 84–89. Nauka, Novosibirsk (1974) (in Russian)
4. Dimitriadou, E., Weingessel, A., Hornik, K.: A combination scheme for fuzzy clustering. J. Pattern Recognit. Artif. Intell. **16**(7), 901–912 (2002)
5. Ghosh, J., Acharya, A.: Cluster ensembles. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **1**, 1–12 (2011)
6. Guenoche, A.: Consensus of partitions: a constructive approach. Adv. Data Anal. Classif. **5**, 215–229 (2011)
7. Meila, M.: Comparing clusterings - an information based distance. J. Multivar. Anal. **98**(5), 873–881 (2007)
8. Mirkin, B.G.: A new approach to the analysis of sociology data. In: Voronov, Y. (ed.) Measurement and Modeling in Sociology, pp. 51–61. Nauka, Novosibirsk (1969) (in Russian)
9. Mirkin, B.G.: Analysis of Categorical Features, 166 pp. Statistika, Moscow (1976) (in Russian)
10. Mirkin, B.: Core Concepts in Data Analysis: Summarization, Correlation, Visualization. Springer, Berlin (2011)
11. Mirkin, B.: Clustering: A Data Recovery Approach. Chapman and Hall, London (2012)
12. Mirkin, B., Muchnik, I.: Geometrical interpretation of clustering scoring functions. In: Mirkin, B. (ed.) Methods for the Analysis of Multivariate Data in Economics, pp. 3–11. Nauka, Novosibirsk (1981) (in Russian)
13. Netlab Neural Network software. http://www.ncrg.aston.ac.uk/netlab/index.php. Accessed 1 Dec (2013)
14. Sevillano, X., Socoro, J.C., Alias, F.: Fuzzy clusterers combination by positional voting for robust document clustering. Procesamiento del lenguaje Nat. **43**, 245–253 (2009)
15. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
16. Topchy, A., Jain, A.K., Punch, W.: A mixture model for clustering ensembles. In Proceedings SIAM International Conference on Data Mining (2004)
17. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. In: Proceedings of the Ninth SIAM International Conference on Data Mining, pp. 211–222 (2009)

Part II
Order and Tree

Single or Multiple Consensus for Linear Orders

Alain Guénoche

Abstract To establish a consensus order, summarizing a profile of linear orders on the same item set is a common problem. It appears in Social Choice Theory, when voters rank candidates in an elective process or in Preference Aggregation, when individuals or criteria put several orders on the items. Often the consensus order is a median order for Kendall's distance, but other definitions, more easily computable, can be used. In the following, we tackle the question of the quality of this summary by a single consensus order. We study the possibility to represent a given profile by several linear orders making a *Multiple Consensus*. We introduce an original criterion to measure the quality of the single or multiple consensus, and so to decide if it is preferable to retain one linear order or to adopt several orders making a better representation. Two applications are described; the first one in Agronomy to select varieties according to yield estimations in several trials and the second one is about the event orders along Jesus, life according to the three Gospels of Mark, Luke, and Matthew.

Keywords Linear orders • Consensus • Median • Preferences • Gospels

1 Introduction

To establish a consensus order, summarizing a set (also denoted as a *profile*) of linear orders on the same *item* set is a common problem. It appears in Social Choice Theory, when voters rank candidates or in Preference Aggregation, when individuals or criteria put several orders on the items. These can be competitive products (wines, perfumes, foods), individuals to select (representative members of a meeting) or to

A. Guénoche (✉)
IML - CNRS 163 Av. de Luminy, 13009 Marseille, France
e-mail: guenoche@iml.univ-mrs.fr

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_12,

reward (students), or some methods dedicated to a specific problem (organizations, algorithms). The comparison between these items can be based on expert opinions or marks given after several tests and also according to quantitative criteria. This gives a set of orders and a decision problem. We will admit that the preferences are linear orders, but the proposed methods can be adapted to the case of ties.

Among equivalent formulations of this problem, we retain here the one of expert rankings. Each judge or expert gives its opinion on a set X of items ($|X| = n$). Each opinion is a linear order or a permutation on X. The set of experts E ($|E| = m$) makes a *profile* $\Pi = \{S_1, \ldots, S_m\}$, in which the m orders are not necessarily different but they all span X (every item is ranked).

In both classical frames, *Social Choice Theory* or *Preference Aggregation*, one tries to establish, from the profile, a collective ordering. For that, a consensus π summarizing preferences is computed. We focus on the case in which this consensus is an element of $\mathcal{S}$, the set of linear orders on X, which is a median for the profile [1,2]. For complexity reasons, other polynomial aggregation strategies can be used, for instance, the famous Borda ranking method (1784) or the Smith and Payne three-cycle elimination algorithm [12]. They can also be used as the median orders in the following.

Orders are binary relations on element pairs of X; x is preferred to y (denoted $x \prec y$) if x is placed before y. Two experts or opinions can be compared counting the number of pairs placed the same way in the orders, that is their number of agreements. The natural distance between orders is the symmetric difference distance on the whole pair set that commonly share the relation. To measure a distance between two experts, it is sufficient to count the number of disagreements, that is Kendall's distance D.

$$\pi = \text{Argmin} \sum_{i=1}^{m} D(\pi, S_i)$$

A median element, relatively to profile Π, is established using score functions with integer values $W_\Pi : \mathcal{S} \to \mathbb{N}$ which must be maximized. It is the same as to minimize the sum of distance values between π and the profile orders. The consensus is a permutation π maximizing W_Π and which is median for Π. When there is a single item to select, it is the first element in this median order. But if there are k items to retain, rather than keeping the k first ranked elements, it could be better to decompose the profile into k classes, to compute a median linear order for each one, and to retain the first ranked item of each order. This is what we call a *Multiple Consensus*.

In this text we recall an algorithm to compute a median order from a given set of linear orders (Sect. 2), then we introduce the Multiple Consensus concept (Sect. 3). In Sect. 4, we detail methods to subdivide a profile, which will be applied to a real selection problem in Agronomy described in Sect. 5; several rapeseed varieties are tested in several trials, each one indicating a yield estimation. These variables, one by place, make preferences and the selection problem highlights the use of Multiple

Consensus. Finally in Sect. 6, we come back to the linear orders of the events along Jesus life, as they are reported in the Gospels of Mark, Luke and Matthew. The three orders are so different that the existence of a single source in questionable. The larger score of a single consensus order will reassure on this origin.

2 Consensus of Linear Orders

Classically, to compute a median order from a linear order profile on X, a pair comparison procedure is applied. First, a table T indexed on $X \times X$ is established:

$$T(x, y) = |\{S \in \Pi \text{ such that } x \prec_S y\}|.$$

So, $T(x, y) + T(y, x) = m$. To this table corresponds a *tournament* (complete directed graph) which is weighted. The arc (x, y) goes from x to y iff $T(x, y) > T(y, x)$; its weight is equal to $w(x, y) = T(x, y) - T(y, x)$ and $w(y, x) = 0$. If $T(x, y) = T(y, x)$ the arc orientation is arbitrary, since both weights are null.

The *remoteness* of a linear order $S = (x_1 \prec x_2 \prec \cdots \prec x_n)$ from a tournament T is equal to the sum of weight of the arcs which disagree with the order.

$$R(S, T) = \sum_{i<j} w(x_j, x_i)$$

When a tournament is transitive, it corresponds to a linear order, which is not necessarily unique (if two consecutive elements are linked by a 0 weighted arc). This linear order is easy to find; it is the decreasing order of the number of dominated vertices in the tournament. Its remoteness to the tournament is equal to 0.

When there is no such permutation, one seeks to reverse a set of arcs having minimum sum of weight to make T transitive. This quantity is equal to the remoteness of the corresponding linear order which is median for the profile defining table T. This problem is well known as the Kemeny Problem (1959). It can be formulated as an integer linear program. For linear orders embedded with Kendall's distance, median linear order computing is NP-Hard [10].

Heuristics to build linear orders close to a tournament (minimizing remoteness) are numerous and we have studied since a long time *Branch and Bound* algorithms to establish an optimal linear order or to enumerate all of them [3, 4, 7]. The first step is to apply a heuristic method [12] providing an upper bound $R_{\max}$ of the remoteness from the tournament. Then, a search tree is developed; nodes correspond to beginning sections of linear orders. Each node is valued by the sum of weights of the reversal arcs it contains. At each step the following operations are performed:

- Find a leaf in the Branch and Bound tree with minimum value,
- Extend this beginning section all the possible ways by an unplaced item if the value of this extended beginning does not overpass $R_{\max}$.

The first leaf containing $n-1$ items determines a linear order π at minimum remoteness from the tournament, so it is a median for profile Π. The weight of this consensus is the sum of the majority opinions on pairs:

$$W_{\Pi}(\pi) = \sum_{x \prec_{\pi} y} T(x,y) - T(y,x).$$

Example 1. Let us consider the profile: $\Pi = \{(1 \prec 2 \prec 3 \prec 4), (1 \prec 3 \prec 2 \prec 4), (1 \prec 2 \prec 4 \prec 3), (3 \prec 1 \prec 4 \prec 2), (2 \prec 3 \prec 4 \prec 1), (4 \prec 2 \prec 3 \prec 1)\}$.

It corresponds to a tournament T; the arc weights are given the following table:

w	1	2	3	4
1	–	2	0	2
2	0	–	2	2
3	0	0	–	2
4	0	0	0	–

Clearly, this tournament is transitive and the weight of the natural order is $W_{\Pi}((1 \prec 2 \prec 3 \prec 4) = 10$.

This kind of consensus is founded if the experts share a collective opinion, a majority of them being very close to. Then, there is a strong consensus, the intensity of which being quantified by function W which counts the majority approvals of pair comparisons. Now, if the consensus is weak, i.e., there are a few majority opinions in the median order, it may be due to several divergent collective opinions within the profile. Added all together they cancel each other out. If we could separate several groups of experts, several different consensus ordering could appear. This decomposition makes sense when talking about notations to students or quantitative criteria corresponding to different variables (price, speed and volume for cars).

This could be important if several items have to be selected. To retain the first ranked elements of a median linear order is not always founded. For instance, if criteria are marks given to students in scientific and literary tests and if two rewards have to be given, it is possible that the first students in sciences are the last ones in literary domain and reciprocally. So the median order will place average students at the two first ranks. But if, on the one hand the scientific disciplines and on the other hand the literary ones are separately considered, the two corresponding median orders will indicate the best students in the two domains.

So we are looking to cluster the experts to put together close linear orders, making appear groups of homogeneous experts sharing, within their group, a strong consensus. There was a first attempt in this direction with the article by Lemaire [11], in which he compares several aggregation procedures and uses the *Nuées dynamiques* algorithm [5] to cluster the profile.

3 Multiple Consensus

To decide if it is preferable to subdivide the expert set and which subdivision is the best one, we define the notion of *generalized score*. Let P^q be a partition of Π in q disjoint classes ($\Pi = \bigcup_{k=1,q} \Pi_k$), and π_k the consensus of each cluster (sub-profile). The generalized score of P^q is the sum of the consensus weights, multiplied by the number of experts in the class.

$$W(P^q) = \sum_{k=1,\ldots,q} |\Pi_k| \times W_{\Pi_k}(\pi_k).$$

This generalized score acts as a ballot quantification. In each class Π_k, the experts vote for the π_k order which is as weighted as they agree with this opinion.

The multiple consensus problem is to maximize W over the set of all the partitions of Π. This problem is not easy, since it requires to fix the optimal value of q and the optimal q-decomposition to evaluate the generalized score. And this latter can be measured after computing the q consensus orders. We denote $\mathcal{W}^q$ the maximum value of W over the set of partitions with q classes.

$$\mathcal{W}^q = \max_{P^q \in \mathcal{P}^q} W(P^q)$$

The consensus of Π in a single class, gives a generalized score $\mathcal{W}^1 = |\Pi| \times W_\Pi(\pi)$. If there exists a q-decomposition of Π such that $\mathcal{W}^q > \mathcal{W}^1$, one can claim that Π contains q opinion groups having their own consensus. Thus, $\mathcal{W}^m$ is the generalized score corresponding to the atomic partition of the profile in which there are only singletons. If score $\mathcal{W}^m$ is the largest one, including $\mathcal{W}^1$, it means that there is no agreement between the m orders in the profile.

Proposition. *The generalized score of the atomic partition Π_0 is $\mathcal{W}^m = m \times \frac{n(n-1)}{2}$.*

Proof. Each linear order is the median order of the sub-profile it makes alone, and its weight is equal to the number of item pairs which are all majority.

Corollary. *Two linear orders admit a single consensus if they agree on more than half the number of item pairs*

Example 2. Coming back to the linear orders of Example 1, the consensus weight of all the orders is 10 and so $\mathcal{W}^1 = 60$. If these opinions are considered as irreconcilable, we get a generalized score $\mathcal{W}^6 = 36$ lower than the single consensus. But if we subdivide Π into two classes Π_1 and Π_2 containing, respectively, the four first orders and the two last ones, we get two tables:

w_1	1	2	3	4
1	–	4	2	4
2	0	–	0	2
3	0	0	–	2
4	0	0	0	–

w_2	1	2	3	4
1	–	0	0	0
2	2	–	2	0
3	2	0	–	0
4	2	0	0	–

The left one designates again the natural order $W_{\Pi_1}(1 \prec 2 \prec 3 \prec 4) = 14$. The Π_2 tournament is also transitive and it admits three median orders (depending on the first item, 2 or 4) having the same weight $W_{\Pi_2}((2 \prec 4 \prec 3 \prec 1) = 8$.

The generalized score of the decomposition $\Pi_1|\Pi_2$ is: $\mathcal{W}^2 = 4\times14+2\times8 = 72$. Consequently, there is a multiple consensus for profile Π.

4 Decomposition Methods for a Profile

To evaluate the generalized score value of a partition, it is necessary to compute first the consensus of the orders in each class Π_i or at least its weight, $W_{\Pi_i}(\pi_i)$. For a linear order profile, the median order problem being NP-Hard, it is impossible to design a polynomial algorithm for an optimal decomposition. For other types of consensus, computable by polynomial heuristics, the problem remains open. We have developed two approximated methods computing first the Kendall's distance D over Π. Then, we apply one method or the other.

4.1 A Hierarchical Method

Since it begins with an optimization on the number of classes, we develop first an algorithm computing a series of partitions having $\{m, (m-1), \ldots, 1\}$ classes. A classical solution consists in an ascending hierarchical procedure (for instance, UPGMA) generating nested partitions (from one to the next, only two classes are joined). For each partition, its generalized score is computed and, finally, the one having the highest value is retained. The atomic partition and also the one with a single class belong to the series. There is no proof of the optimality of this best computed partition over the set of them all on Π.

4.2 Partitioning by the Fusion–Transfer Algorithm

We have adapted an optimization procedure for graph partitioning to a valued function on $\Pi \times \Pi$ to build directly a partition of Π. First, a similarity index is defined, depending on one parameter. Let $\overline{D}$ be the average value of D, $D_{\max}$ be its maximum value and α a parameter in $[0, (D_{\max}/\overline{D}]$. The similarity function $B : \Pi \times \Pi \to \mathbb{R}$ is defined as

$$B(S_i, S_j) = \alpha \times \overline{D} - D(S_i, S_j).$$

The aim of this clique partitioning algorithm is to maximize the sum of the joined pairs values within the classes. For a partition P^q of Π in q disjoint classes ($\Pi = \bigcup_{k=1,q} \Pi_k$), the value of this partition is given by:

$$\mathcal{B}(P^q) = \sum_{k=1}^{q} \sum_{S_i, S_j \in \Pi_k} B(S_i, S_j).$$

When $\alpha > (D_{\max}/\overline{D})$, all the values of B are positive or null and the maximization gives the partition with a single class; when $\alpha = 0$ they are all negative and one gets the atomic partition. Between them, the B values are either positive or negative and the number of classes is automatically determined by the algorithm.

To maximize the sum of the intra-class pair values, positive or negative, is a problem which arises in *Graph Partitioning* when communities are searched, optimizing a *modularity* function. We use again our Fusion–Transfer algorithm, defined for the consensus partition problem [8].

- The first part, *Fusion*, is a hierarchical ascending method. Starting from the atomic partition Π_0, at each step the two classes maximizing the score value of the resulting partition are merged. The process stops when there is no fusion increasing $\mathcal{B}$. It leads to partition $P^q = (\Pi_1, \ldots, \Pi_q)$ such that any partition obtained from P^q merging two classes has a lower $\mathcal{B}$ score.
- In the second part, *Transfer*, the weight of the assignment of any element to any class is computed. Let $A(i,k) = \sum_{S_j \in \Pi_k} B(S_i, S_j)$. If $S_i \in \Pi_k$, $A(i,k)$ is the contribution of S_i to its own class, and also to $\mathcal{B}(P^q)$. Otherwise, the $A(i,k')$ value corresponds to a possible assignment to class $\Pi_{k'}$. The difference $A(i,k') - A(i,k)$ is the $\mathcal{B}$ variation resulting from the transfer of S_i from class Π_k to class $\Pi_{k'}$. Our procedure, consists in moving at each step the element maximizing this difference. Order S_i is assigned either to class $\Pi_{k'}$ if $A(i,k') \geq 0$ or a new singleton class is created. In this latter case, S_i has a null contribution to $\mathcal{B}$, increasing the criterion value. We have implemented this algorithm, making a table A, indexed on Π and on the classes of the running partition. The transfer procedure stops when each item has a non negative contribution to its class which is larger than or equal to its contribution to any other class.

4.3 The Fusion–Transfer Algorithm

- **Hierarchical procedure**
  - Start from π_0
  - Compute the variation of score due to the fusion of any pair ($B(S_i, S_j)$)
  - While score $\mathcal{B}$ increases
    - Join the two classes giving the maximum variation
    - Update the fusion gains of the new class with the remaining classes
- **Transfer procedure**
  - Compute the weight of any element in any class (Table A)

- Memorize the best class (maximum value) for any element
- While there exists an element whose weight in its class is not maximum,
  - put it into the class where its contribution is maximum, if ≥ 0, otherwise make it a singleton;
  - update the weights of the elements in both modified classes

The Fusion part is like a hierarchical algorithm and so is in $O(m^3)$; the transfer part is in $O(mq)$ at each transfer and much faster than the previous part. This algorithm is fast enough to subdivide large profiles with hundreds of orders in less than 1 min. Probably, there is no problem with such size for expert opinions, but it could be possible for criteria orderings.

5 Agronomical Selection

Many years ago, I have participated in the analysis of agronomical data [9]. The question was to select promising rapeseed (colza) varieties to put them on the marketplace or to try to improve them again. These varieties have been tested along multiple trials in different INRA stations, where the average yield of each one has been estimated. It was the only criterion. Each trial provides a yield value distribution and they were very different. In front of these fuzzy measures, we adopt an ordinal approach, transforming the distributions of yield into orders, and counting the number of times (trials) one variety has a higher yield than another.

Example 3. To illustrate the profile decomposition interest, I go back to a subset of these data considering the linear orders established by 22 INRA trials on 6 rapeseed varieties. The resulting majority tournament is:

	1	2	3	4	5	6
1	–	0	0	0	0	0
2	4	–	4	0	0	4
3	4	0	–	0	0	4
4	10	0	4	–	2	6
5	14	4	8	0	–	6
6	0	0	0	0	0	–

It is transitive and provides two median orders: $(4 \prec 5 \prec 2 \prec 3 \prec \{1, 6\})$ having weight 74; hence the generalized score of a single consensus is $\mathcal{W}^1 = 22 \times 74 = 1628$.

But the hierarchical method applied to Kendall's distance between the 22 linear orders leads to 2 classes with 13 and 9 trials, respectively. The two corresponding tournaments are:

	1	2	3	4	5	6	1	2	3	4	5	6
1	–	0	0	0	0	1	–	5	5	0	0	0
2	9	–	5	9	1	11	0	–	0	0	0	0
3	9	0	–	5	0	9	0	1	–	0	0	0
4	3	0	0	–	0	3	7	9	9	–	3	3
5	11	0	3	1	–	7	3	5	5	0	–	0
6	0	0	0	0	0	–	1	7	5	0	1	–

The left one is transitive and corresponds to a unique median order ($2 \prec 5 \prec 3 \prec 4 \prec 1 \prec 6$) with weight 87 while the right one is also transitive for the order ($4 \prec 6 \prec 5 \prec 1 \prec 3 \prec 2$) having weight 69. The generalized score of this bipartition is larger than $\mathcal{W}^1$, since $\mathcal{W}^2 = 13 \times 87 + 9 \times 69 = 1131 + 621 = 1752$. With the second method and $\alpha = 1$, we find back the same decomposition into two classes and, with a larger value of α, we obtain three classes or more with a lower score value (1463).

Consequently, if agronomists decide to select two varieties, with a single median order they will retain 4 and 5. But the decomposition improving the generalized score suggests to retain 2 and 4; the two orders may correspond to different soils or climatic conditions.

6 Comparing the Event Orders in Jesus, Life

Long time ago, I met Louis Frey, who spent several years to compare the Synoptic Gospels of Mark, Luke and Matthew. Observing that the narrations are as divergent as they agree, he interested himself in the orders of the events in Jesus, life according to the three relations, the fourth from John being not comparable. He wrote an amazing book containing all necessary data [6].

His first task was to enumerate all the events and to label them, corresponding, for instance, to speeches (the prediction of the birth of John the Baptist, the straw and the beam), to maxims (The Talion Law), to miracles or reported facts (the leper's healing) by one or the other Evangelists. These events denoted *blocks* and are precisely defined according to the number of consecutive verses that can be shared by two or three narrations or specific to a single one. An annex document ranks, in the Marc order, the block positions; so it is very easy to establish the permutations.

Since the narrations are far to agree, I wonder if there is a unique or multiple consensus from these orders. Even common blocks are not sorted in the same way. At the beginning, there are 428 blocks, but only 87 of them are present in the three Gospels which, respectively, contain 202, 271 and 297 blocks. This means that many of them are unique.

The three permutations are displayed hereafter:

Mark: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87
Luke: 3 2 4 44 5 6 7 11 38 39 9 10 18 8 12 13 14 15 16 17 19 28 1 23 24 25 26 27 29 22 31 32 33 34 35 36 37 40 41 42 43 45 46 47 50 51 52 53 54 55 56 58 57 59 65 77 20 48 49 21 30 67 61 62 60 73 85 63 64 66 68 70 71 82 72 74 75 76 78 79 80 81 83 84 86 87 69
Matthew: 3 2 4 5 6 7 8 11 61 26 62 28 9 12 10 31 32 33 34 13 35 36 37 19 40 41 42 83 27 59 1 14 15 16 17 18 20 21 22 23 29 24 25 30 38 39 43 44 45 46 47 48 49 50 51 52 53 54 73 55 56 58 64 60 63 65 66 67 68 69 57 70 71 72 74 75 76 77 79 78 80 81 82 84 85 86 87

To compare these permutations, Frey makes several diagrams linking the identical blocks and showing numerous crossings. He also measured Kendall's distance between them and assesses that Mark's Gospel is between the two others of Luke and Matthew. Using a methodology coming from genome comparison, I computed the longest common chains between these orders; these are series of blocks, not necessarily consecutive but placed in the same order. Between Mark and Luke, 63 blocks are sorted in the same order; between Mark and Matthew there are 60 blocks, but comparing Luke and Matthew, only 47 blocks are found. Looking for a longest common chain to the three narrations, only 45 blocks can be found, just a little more than half the common ones: this is one of the longest chains:

2 4 5 6 7 11 28 31 32 33 34 35 36 37 40 41 42 43 45 46 47 50 51 52 53 54 55 56 58 60 63 66 68 70 71 72 74 75 76 79 80 81 84 86 87.

As for Kendall's distance values, these figures prove that Mark is closer to Luke and Matthew than the later evangelists are. Consequently we may think, according to Frey, that Mark's Gospel was known by the two other evangelists. Although Mark wasn't a of Jesus disciple, while Luke and Matthew were, recent studies confirmed that Luke used Mark's Gospel as one of his sources. But I am more concerned by the question "Is there sufficient common information to assess the uniqueness of the source" rather than "Who wrote his Gospel first."

Looking to consensus weights, the generalized score of the three evangelists in one class is much larger ($\mathcal{W}^1 = 29217$) than the generalized score of these same evangelists considered as independent ($\mathcal{W}^3 = 11223$). This is a new argument to claim the unicity of the origin of the Gospels, even if strong divergences remain unexplained, may be due to the author's creativity or to the copyist's extravagance.

7 Conclusions

We have described a simple method, based on the original generalized score function, to realize a partitioning of a set of orders. It is an ordinal method extending the consensus of a linear order profile, to a multiple consensus to improve the weight of the collective opinion.

The optimal partitioning of the profile is not guaranteed, and for large-size problems, the computed median orders could be sub-optimal. Nevertheless, if a generalized score of a partition in several classes is higher than the one of the whole profile, one can claim this profile is not homogeneous and contains several different opinions. In that case a multiple consensus provides a better summarizing of them all.

This decomposition can be very useful in case of selection of several items in a multicriteria selecting process.

References

1. Barthélemy, J.P., Leclerc, B.: The median procedure for partitions. In: Cox, I.J., Hansen, P., Julesz, B. (eds.) Partitioning Data Sets. DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences, vol. 19, pp. 3–34. American Mathematical Society, Providence, Rhode Island (1995)
2. Barthélemy, J.P., Monjardet, B.: The median procedure in cluster analysis and social choice theory. Math. Soc. Sci. **1**, 235–267 (1981)
3. Barthélemy, J.P., Guénoche, A., Hudry, O.: Median linear orders: Heuristics and branch and bound algorithm. Eur. J. Oper. Res. **42**, 555–579 (1989)
4. Charon, I., Guénoche, A., Hudry, O., Woirgard, F.: A Bonsaï Branch and Bound method applied to voting theory. In: Diday, E., et al. (eds.) Proceedings of "Ordinal and Symbolic Data Analysis" (OSDA'95), pp. 309–318. Springer, Berlin (1996)
5. Diday, E.: Une nouvelle méthode en classification automatique et reconnaissance des formes. Rev. Stat. Appl. **19**, 2 (1971)
6. Frey, L.: Analyse ordinale des évangiles synoptiques. Gauthier-Villars, Paris (1972)
7. Guénoche, A.: Un algorithme pour pallier l'effet Condorcet. R.A.I.R.O. Rech. Opér. **11**(1), 77–83 (1977)
8. Guénoche, A.: Consensus of partitions: a constructive approach. Adv. Data Anal. Classif. **5**(3), 215–229 (2011)
9. Guénoche, A., Vandeputte-Riboud, B., Denis, J.-B.: Selecting varieties using a series of trials and a combinatorial ordering method. Agronomie **14**, 363–375 (1994)
10. Hudry, O.: Recherche d'ordres médians: complexité, algorithmique et problèmes combinatoires. Thèse de l'ENST, Paris (1989)
11. Lemaire, J.: Agrégation typologique de données de préférence. Math. Sci. Hum. **58**, 31–50 (1977)
12. Smith, A.F.M., Payne, C.D.: An algorithm for determining Slater's i and all nearest adjoining orders. Br. J. Math. Stat. Psychol. **27**, 49–52 (1974)

Choice Functions on Tree Quasi-Orders

F.R. McMorris and R.C. Powers

This paper is dedicated to Boris Mirkin on the occasion of his 70th birthday

Abstract The domain of social choice functions is extended to tree quasi-orders, and versions of the theorems of Arrow, Muller–Satterthwaite, and Gibbard–Satterthwaite are proved in this setting.

Keywords Choice function • Consensus function • Tree quasi-order • Strategy-proof

1 Introduction

Social welfare functions defined on various types of preference relations have been, and continue to be, well studied. (cf. [1, 6, 9, 13]) Under this formalism, functions defined on other discrete structures such as tree-like hypergraphs have been called consensus functions. Consensus functions have been extensively studied and applied in classification theory, systematic biology, and other areas where aggregation methods might be used [5]. For example, a direct version of Arrow's theorem for consensus functions defined on tree quasi-orders was proved in [11] and an analog

F.R. McMorris (✉)
Department of Applied Mathematics, Illinois Institute of Technology, Chicgo, IL 60616, USA

Department of Mathematics, University of Louisville, Louisville, KY 40292, USA
e-mail: mcmorris@iit.edu

R.C. Powers
Department of Mathematics, University of Louisville, Louisville, KY 40292, USA
e-mail: robert.powers@louisville.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_13,

of Wilson's theorem for tree quasi-orders was proved in [12]. In the present paper we consider choice functions on tree quasi-orders and prove analogs of the theorems of Arrow, Muller–Satterthwaite, and Gibbard–Satterthwaite.

2 Definitions and Axioms

Let A be a finite set of alternatives with $|A| \geq 3$, and ρ a binary relation on A (i.e., a subset of $A \times A$). We will write $x\rho y$ instead of $(x, y) \in \rho$, $\neg(x\rho y)$ if $(x, y) \notin \rho$, and $x\rho^* y$ if $x\rho y$ and $\neg(y\rho x)$. The set of maximal elements of ρ is $\text{Max}(\rho) = \{x \in A : \neg(y\rho^* x) \text{ for all } y \in A\}$. If $X \subseteq A$, then $\rho|_X$ denotes the restriction $\rho \cap X \times X$. Recall that a binary relation that is reflexive and transitive is a *quasi-order* and a complete quasi-order is a *weak order*. In most models of social choice, preference relations are required to be weak orders, whereby an individual either strictly prefers one alternative to another or is indifferent to the two alternatives. The simplest type of weak order (not allowing indifference) is a *linear order*, which is a complete, transitive, anti-symmetric binary relation. To generalize away from the requirement that a preference relation be complete, and allow alternatives to be declared incomparable, we consider tree quasi-orders. A *tree quasi-order* is a quasi-order τ that satisfies the tree condition:

$$z\tau y \text{ and } z\tau x \Rightarrow x\tau y \text{ or } y\tau x \text{ for any } x, y, z \in A.$$

Thus if an individual's preference relation is modeled as a tree quasi-order, then a comparison is required between x and y when both are less preferred than some other alternative z; otherwise, it may be possible for x and y to be incomparable.

Let $\mathscr{L}$, $\mathscr{W}$, and $\mathscr{T}$ be the set of linear orders, weak orders, and tree quasi-orders on A, respectively. Clearly, $\mathscr{L} \subseteq \mathscr{W} \subseteq \mathscr{T}$.

Social welfare functions and social choice functions are two standard types of functions encountered in mathematical social sciences when considering a society of voters each having declared a preference relation. The current terminology used when the domain extends away from linear or weak orders is the following. Let k be an integer where $k \geq 2$ and $K = \{1, \ldots, k\}$. A *consensus function* on $\mathscr{T}$ is a mapping $f : \mathscr{T}^k \to \mathscr{T}$ while a *choice function* on $\mathscr{T}$ is a mapping $g : \mathscr{T}^k \to A$. Elements of $\mathscr{T}^k$ are called *profiles* and denoted by $P = (\tau_1, \ldots, \tau_k)$, $P' = (\tau'_1, \ldots, \tau'_k)$, etc. For any profile P and $X \subseteq A$, set $P|_X = (\tau_1|_X, \ldots, \tau_k|_X)$.

In [11] a direct analog of Arrow's theorem [2] was established for consensus functions on tree quasi-orders. In order to contrast the axioms for choice functions with those for consensus functions, we recall these standard axioms.

Let $f : \mathscr{T}^k \to \mathscr{T}$ be a consensus function.

P: f satisfies the *Pareto* condition if, for all $x, y \in A$ and profiles $P = (\tau_1, \ldots, \tau_k)$,

$$x\tau_i^* y \text{ for all } i \in K \Rightarrow xf(P)^* y.$$

IIA: f satisfies *Independence of Irrelevant Alternatives* if, for all $x, y \in A$ and profiles $P, P' \in \mathscr{T}^k$,

$$P|_{\{x,y\}} = P'|_{\{x,y\}} \Rightarrow f(P)|_{\{x,y\}} = f(P')|_{\{x,y\}}.$$

D: f is a *Dictatorship* if there exists $j \in K$ such that for any profile $P = (\tau_1, \ldots, \tau_k)$ and $x, y \in A$,

$$x\tau_j^* y \;\Rightarrow\; xf(P)^* y.$$

We can now state the result in [11], whose proof follows along standard lines with some modifications needed to account for the lack of completeness in tree quasi-orders.

Theorem 1. *If $|A| \geq 3$, a consensus function on $\mathscr{T}$ that satisfies IIA and P must be a dictatorship.*

In [12], analogous to results of [10, 17, 18], we investigated consensus functions on $\mathscr{T}$ that satisfied IIA but not P. Replacing P with two simple profile conditions we showed that if f satisfies IIA and these two conditions, then the symmetric part of f is oligarchical and the asymmetric part of f is either trivial or quasi-oligarchical.

We now turn our attention to choice functions on $\mathscr{T}$, the main topic of this paper. As in [3] we use an asterisk to distinguish the choice axioms from the consensus axioms. Let $g : \mathscr{T}^k \to A$ be a choice function, and let $g(\mathscr{T}^k) = \{x \in A : g(P) = x \text{ for some } P \in \mathscr{T}^k\}$.

P*: g satisfies the *Pareto* condition if, for all $x, y \in A$ and profiles $P = (\tau_1, \ldots, \tau_k)$,

$$x \in g(\mathscr{T}^k) \text{ and } x\tau_i^* y \text{ for all } i \in K \;\Rightarrow g(P) \neq y.$$

IIA*: g satisfies *Independence* if, for all $x, y \in A$ with $x \neq y$ and profiles $P, P' \in \mathscr{T}^k$,

$$g(P) = x \text{ and } P|_{\{x,y\}} = P'|_{\{x,y\}} \;\Rightarrow\; g(P') \neq y.$$

D*: g is a *Dictatorship* if there exists a $j \in K$, called a *g-dictator*, such that for any $P = (\tau_1, \ldots, \tau_k) \in \mathscr{T}^k$, $\neg(x\tau_j^* g(P))$ holds for all $x \in g(\mathscr{T}^k)$.

Note that $\neg(x\tau_j^* g(P))$ for all $x \in g(\mathscr{T}^k)$ can be stated as $g(P) \in \mathrm{Max}(\tau_j|_{g(\mathscr{T}^k)})$.

We will need the following theorem of Arrow for choice functions on linear orders [3].

Theorem 2. *Let g be a choice function $g : \mathscr{L}^k \to A$ with $|g(\mathscr{L}^k)| \geq 3$. If g satisfies IIA* and P*, then g must be a dictatorship on $\mathscr{L}$.*

3 Main Result

Our main result will be to extend Theorem 2 from linear orders to tree quasi-orders.

Theorem 3. *If $g : \mathscr{T}^k \to A$ satisfies IIA*, P*, and $|g(\mathscr{T}^k)| \geq 3$, then there is a g-dictator.*

Proof. Assume g is a choice function on $\mathscr{T}$ that satisfies IIA* and P*, with $|g(\mathscr{T}^k)| \geq 3$. Using P* it can be easily shown that $g(\mathscr{L}^k) = g(\mathscr{T}^k)$ and so $|g(\mathscr{L}^k)| \geq 3$. Now g restricted to $\mathscr{L}^k$ satisfies IIA* and P* and so, by Theorem 2, this restricted map is a dictatorship. Without loss of generality assume $j = 1$ to be the dictator when g is operating on $\mathscr{L}^k$. Our goal is to show that 1 is a dictator on the entire domain $\mathscr{T}^k$.

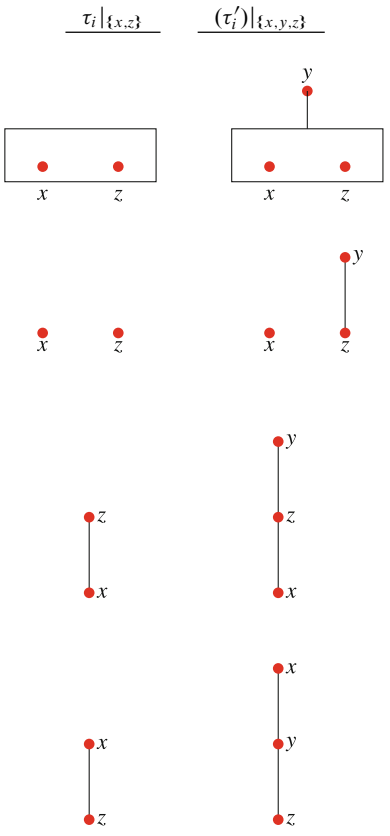


Fig. 1 Placing y in P' with respect to x and z.

Suppose $j = 1$ is not a g-dictator. This means that there exist $P = (\tau_1, \ldots, \tau_k) \in \mathscr{T}^k$ such that $g(P) = z$ and $x\tau_1^* z$ for some $x \in g(\mathscr{T}^k)$. We will argue to a contradiction by using a series of modifications to the profile P. Let $y \in g(\mathscr{T}^k) \setminus \{x, z\}$ and define $P' = (\tau_1', \ldots, \tau_k') \in \mathscr{T}^k$ based on the four possibilities for $\tau_i|_{\{x,z\}}$. Figure 1 shows how y is inserted when forming τ_i'. We also require, for each $i \in K$, that $x(\tau_i')^* w$, $y(\tau_i')^* w$, and $z(\tau_i')^* w$ for every $w \in A \setminus \{x, y, z\}$. It follows from P* that $g(P') \neq w$ for all $w \in A \setminus \{x, y, z\}$. Since $P'|_{\{x,z\}} = P|_{\{x,z\}}$ and $g(P) = z$ it follows from IIA* that $g(P') \neq x$. Notice that $y(\tau_i')^* z$ for all $i \in K$. Therefore, by P*, $g(P') \neq z$. Therefore, $g(P') = y$.

Our next step is constructing another profile $P'' = (\tau_1'', \ldots, \tau_k'')$ that satisfies the relationships depicted in Fig. 2 based on the three possibilities for $(\tau_i')|_{\{x,y\}}$.

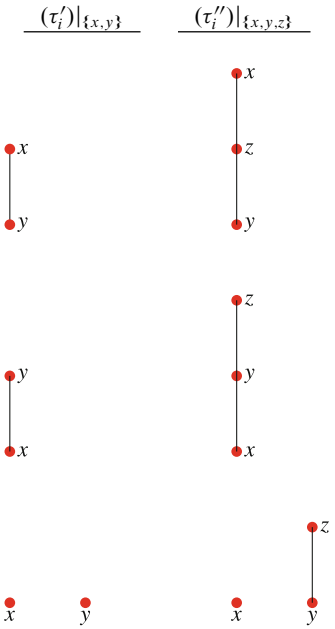


Fig. 2 Placing z in P'' with respect to x and y.

We also require $x(\tau_i'')^* w$, $y(\tau_i'')^* w$, and $z(\tau_i'')^* w$ for every $w \in A \setminus \{x, y, z\}$. Since $P''|_{\{x,y\}} = P'|_{\{x,y\}}$ and $g(P') = y$ it follows from IIA* that $g(P'') \neq x$. Since $z(\tau_i'')^* y$ for all $i \in K$, P* implies that $g(P'') \neq y$. Since $z(\tau_i'')^* w$ for all $i \in K$ and any $w \in A \setminus \{x, y, z\}$ it follows again from P* that $g(P'') \neq w$ for all $w \in A \setminus \{x, y, z\}$. Thus, $g(P'') = z$.

The third step is to construct a profile $P^3 = (\tau_1^3, \ldots, \tau_k^3)$ that satisfies the conditions in Fig. 3

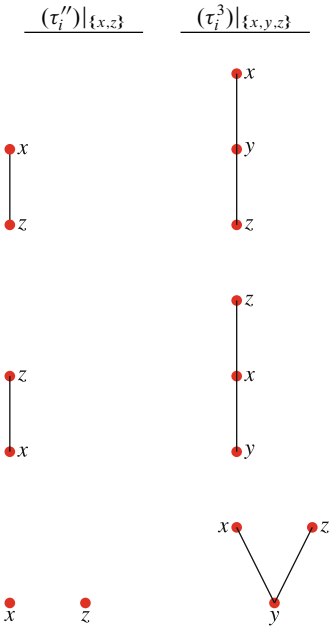


Fig. 3 Placing y in P^3 with respect to x and z.

As in the previous constructions, we will require $x(\tau_i^3)^*w$, $y(\tau_i^3)^*w$, and $z(\tau_i^3)^*w$ for every $w \in A \setminus \{x, y, z\}$. Observe that $x(\tau_i^3)^*y$ for all $i \in K$ and so $g(P^3) \neq y$ by P*. Also we have that $x(\tau_i^3)^*w$ for all $i \in K$ and $w \in A \setminus \{x, y, z\}$ so P* implies $g(P^3) \neq w$ for all $w \in A \setminus \{x, y, z\}$. Since $P^3|_{\{x,z\}} = P''|_{\{x,z\}}$ and $g(P'') = z$ it follows from IIA* that $g(P^3) \neq x$ and therefore $g(P^3) = z$.

Finally construct a fourth profile P^4 such that $P^4|\{y, z\} = P^3|\{y, z\}$ and all other elements of A form a linear order strictly below y and z in every τ_i^3. Since either $y(\tau_i^3)^*z$ or $z(\tau_i^3)^*y$ for all $i \in K$ it follows that $P^4 \in \mathscr{L}^k$. Following what happens to $j = 1$ when the profiles are modified we get $y(\tau_1^4)^*z$. Since g restricted to $\mathscr{L}^k$ has dictator $j = 1$ it follows that $g(P^4) = y$. Now $P^4|_{\{y,z\}} = P^3|_{\{y,z\}}$ and $g(P^4) = y$ so IIA* implies $g(P^3) \neq z$, a contradiction. □

The role of completeness is subtle when studying Arrovian social choice functions. The following example shows that if the domain is restricted in a specified way, then it is possible to have a function satisfying IIA*, P*, and non-dictatorship.

Example. Suppose $A = \{x, y, z\}$, $\tau = \delta \cup \{(y, z)\}$, and $\tau' = \delta \cup \{(x, y), (x, z)\}$. The quasi-orders τ and τ' are shown in Fig. 4.

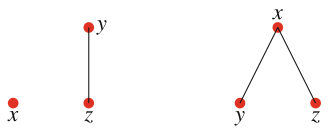


Fig. 4 The quasi-orders τ and τ'

Let

$$D = \mathscr{W} \cup \{\tau, \tau'\}.$$

Define $g : D^2 \to A$ as follows. For any profile $P = (R_1, R_2) \in D^2$,

$$g(P) = \begin{cases} z & \text{if } zR_1^*y,\ zR_1^*x, \text{ and } R_2 \neq \tau \\ y & \text{if } yR_1z,\ yR_1^*x, \text{ and } R_2 \neq \tau \\ x & \text{otherwise.} \end{cases}$$

The choice function g satisfies IIA*, P*, and is not a dictatorship.

Here is an argument for why g satisfies IIA*. Suppose $P = (R_1, R_2)$ and $P' = (R_1', R_2')$ are two profiles. If $P|_{\{y,z\}} = P'|_{\{y,z\}}$, then $zR_1^*y \Leftrightarrow z(R_1')^*y$ and so $f(P) = z$ implies that $g(P') \neq y$.

If $P|_{\{x,z\}} = P'|_{\{x,z\}}$ and $g(P) = z$, then zR_1^*x, and $R_2 \neq \tau$. Now zR_1^*x, and $R_2 \neq \tau$ implies that $z(R_1')^*x$, and $R_2' \neq \tau$. Observe that $z(R_1')^*x$ implies that $R_1' \in \mathscr{W}$ and so either $yR_1'z$ or $z(R_1')^*y$. If $yR_1'z$, then $z(R_1')^*y$ implies that $z(R_1')^*x$ and we get $g(P') = y$. If $z(R_1')^*y$, then we get $g(P') = z$. In either case, $g(P') \neq x$.

If $P|_{\{x,y\}} = P'|_{\{x,y\}}$ and $g(P) = y$, then we get $y(R_1')^*x$, and $R_2' \neq \tau$. Now $y(R_1')^*x$ implies that $R_1' \in \mathscr{W}$ and so either $yR_1'z$ or $z(R_1')^*y$. If $yR_1'z$, then we get $g(P') = y$. If $z(R_1')^*y$, then $y(R_1')^*x$ implies that $z(R_1')^*x$ and it follows that $f(P') = z$. In either case, $g(P') \neq x$. Hence g satisfies IIA*.

Observe that $g(P) \in \mathrm{Max}(R_1) \cup \mathrm{Max}(R_2)$ for any profile $P = (R_1, R_2) \in D^2$. This implies that g satisfies P*.

To see why g is not a dictatorship, consider the profile $P = (R_1, R_2)$ where $R_1 = \delta \cup \{(y, x), (x, z), (y, z)\}$ and $R_2 = \tau$. Since $R_2 = \tau$ it follows that $g(P) = x$. Now $yR_1^*g(P)$ with $y \in g(D)$ implies that 1 is not a g-dictator. Since it is clear that 2 is not a g-dictator it follows that g is not a dictatorship.

4 Monotone and Strategy-Proof Choice Functions

There are several other important theorems in addition to Arrow's theorem on social choice functions that are now considered classics. Among these are the theorems of Muller and Satterthwaite [14], and Gibbard–Satterthwaite [8, 16] that present other reasonable axioms on a social choice function that lead to dictatorships. There

are many expositions of these theorems, a nice one being [15]. In this section we establish versions of these theorems for tree quasi-orders. Let $g : \mathscr{T}^k \to A$ be a choice function.

M*: g is *Monotone* if for any $x \in A$ and profiles $P = (\tau_1, \ldots, \tau_k)$, $P' = (\tau_1', \ldots, \tau_k')$

$$g(P) = x \Rightarrow g(P') = x$$

whenever $\{i : x\tau_i^* y\} \subseteq \{i : x(\tau_i')^* y\}$ for all $y \in A$ with $x \neq y$.

The next theorem is our analog for the Muller–Satterthwaite theorem [14].

Theorem 4. *If g is a choice function $g : \mathscr{T}^k \to A$ that satisfies M* and $|g(\mathscr{T}^k)| \geq 3$, then there is a g-dictator.*

Proof. Assume g satisfies the conditions M* and $|g(\mathscr{T}^k)| \geq 3$. We will prove g satisfies IIA* and P*, so by Theorem 3 the result follows. To show that g satisfies IIA* let $P = (\tau_1, \ldots, \tau_k)$ and $P' = (\tau_1', \ldots, \tau_k')$ be profiles such that $P|_{\{x,y\}} = P'|_{\{x,y\}}$. Suppose $g(P) = x$. We must show $g(P') \neq y$.

Construct $P'' = (\tau_1'', \ldots, \tau_k'')$ where each τ_i'' has $x(\tau_i'')^* z$ and $y(\tau_i'')^* z$ for all $z \in A$ with $z \notin \{x, y\}$ and also $P''|_{\{x,y\}} = P|_{\{x,y\}} (= P'|_{\{x,y\}})$. Now $g(P) = x$ and $\{i : x\tau_i^* z\} \subseteq \{i : x(\tau_i'')^* z\}$ for all $z \neq x$. So M* gives $g(P'') = x$. Using the same reasoning we get if $g(P') = y$, then $g(P'') = y$. Therefore we must have $g(P') \neq y$.

To show g satisfies P*, let $x, y \in A$ with $x \in g(\mathscr{T}^k)$ and let $P = (\tau_1, \ldots, \tau_k)$ be a profile such that $x\tau_i^* y$ for all $i \in K$. We must show $g(P) \neq y$. Since $x \in g(\mathscr{T}^k)$, there exists a profile $P' = (\tau_1', \ldots, \tau_k')$ such that $x = g(P')$. Let $P'' = (\tau_1'', \ldots, \tau_k'')$ where each τ_i'' is a weak order with $x(\tau_i'')^* z$ for all $z \neq x$. Then $\{i : x(\tau_i')^* z\} \subseteq \{i : x(\tau_i'')^* z\}$ for all $z \neq x$, so M* implies $g(P'') = x$. Since $P''|_{\{x,y\}} = P|_{\{x,y\}}$ and g satisfies IIA* from the above, we have $g(P) \neq y$. □

SP*: The choice function g is *Strategy-Proof* if for every profile $P = (\tau_1, \ldots, \tau_k)$, $\tau \in \mathscr{T}$ and every $i \in K$, $\neg(g(\tau_1, \ldots, \tau_{i-1}, \tau, \tau_{i+1}, \ldots, \tau_k)\tau_i^* g(P))$.

Theorem 5 is the Gibbard–Satterthwaite theorem for weak orders.

Theorem 5. *If g is a choice function on weak orders, $g : \mathscr{W}^k \to A$ that satisfies SP* and $|g(\mathscr{W}^k)| \geq 3$, then there is a g-dictator.*

This theorem can also be extended easily to tree quasi-orders, following a proof outline similar to that found in [7]. We note that Campbell [4] has proved something even more general so that this theorem holds for any domain that contains the linear orders. Nevertheless we include a proof of Theorem 6.

Theorem 6. *If g is a choice function on $\mathscr{T}$ that satisfies SP* and $|g(\mathscr{T}^k)| \geq 3$, then there is a g-dictator.*

Proof. Assume g is a choice function on $\mathscr{T}$ that satisfies SP* and $|g(\mathscr{T}^k)| \geq 3$. We first show $g(\mathscr{W}^k) = g(\mathscr{T}^k)$. Clearly $g(\mathscr{W}^k) \subseteq g(\mathscr{T}^k)$. Assume $x \in g(\mathscr{T}^k)$ but

$x \notin g(\mathscr{W}^k)$. Then $x = g(P)$ for some $P = (\tau_1, \ldots, \tau_k) \in \mathscr{T}^k$. Now $x \notin g(\mathscr{W}^k)$ implies that for a particular $P' = (\omega_1, \ldots, \omega_k) \in \mathscr{W}^k$ with $\{x\} = \text{Max}(\omega_i)$ for all $i \in K$, we have $g(P') \neq x$. If $f(\omega_1, \tau_2, \ldots, \tau_k) \neq x$, then

$$f(\tau_1, \ldots, \tau_k)\omega_1^* f(\omega_1, \tau_2, \ldots, \tau_k)$$

contrary to the fact that g satisfies SP*. If $f(\omega_1, \tau_2, \ldots, \tau_k) = x$, then there exists a smallest integer $j \geq 2$ such that $g(\omega_1, \ldots, \omega_{j-1}, \tau_j, \ldots, \tau_k) = x$, but $g(\omega_1, \ldots, \omega_j, \tau_{j+1}, \ldots, \tau_k) = y \neq x$. Since $x\omega_j^* y$, this again leads to a contradiction to that fact that g satisfies SP*. So we must have $x \in g(\mathscr{W}^k)$.

Let $g^\star = g|_{\mathscr{W}^k}$. Since $g^\star$ satisfies SP* and $|g^\star(\mathscr{W}^k)| \geq 3$, by Theorem 5 there is a $g^\star$-dictator. Without loss of generality suppose $i = 1$ is the $g^\star$-dictator. We will show that g is a dictatorship with $i = 1$ being the g-dictator. Let $P = (\tau_1, \ldots, \tau_k)$ be a profile in $\mathscr{T}^k$. We must show $g(P) \in \text{Max}(\tau_1)$.

Construct k weak orders $\omega_1, \ldots, \omega_k$ that satisfy the following: $\text{Max}(\omega_1) = \text{Max}(\tau_1)$ and $\text{Max}(\omega_i) = A \setminus \text{Max}(\tau_1)$ for all $i \neq 1$. Consider the profiles $P_0 = P = (\tau_1, \ldots, \tau_k)$ and for each $i \in K$ let $P_i = (\omega_1, \ldots, \omega_i, \tau_{i+1}, \ldots, \tau_k)$. So $g^\star(P_k) = g(P_k) \in \text{Max}(\tau_1) = \text{Max}(\omega_1)$. Since $g(P_k) \in \text{Max}(\tau_1)$, there is a smallest j such that $g(P_j) \in \text{Max}(\tau_1)$. If $j = 0$, then $g(P_0) \in \text{Max}(\tau_1)$ which is what we want to prove. If $j = 1$, then $g(P_1)(\tau_1)^* g(P)$ which contradicts SP*. If $j > 1$, then $g(P_{j-1})(\omega_j)^* g(P_j)$ which also contradicts SP*, and the proof is complete. □

References

1. Aleskerov, F.: Arrovian Aggregation Models. Kluwer, Boston (1999)
2. Arrow, K.J.: Social Choice and Individual Values. Wiley, New York (1951)
3. Beja, A.: Arrow and Gibbard-Satterthwaite revisited. Math. Soc. Sci. **25**, 281–286 (1993)
4. Campbell, D.E.: Equity, Efficiency, and Social Choice. Clarendon Press, Oxford (1992)
5. Day, W.H.E., McMorris, F.R.: Axiomatic Consensus Theory in Group Choice and Biomathematics. Frontiers in Applied Mathematics. SIAM, Philadelphia (2003)
6. Gaertner, W.: A Primer in Social Choice Theory. Oxford University Press, Oxford (2006)
7. Gärdenfors, P.: A concise proof of theorem on manipulation of social choice functions. Public Choice **32**, 137–142 (1977)
8. Gibbard, A.: Manipulation of voting schemes: a general result. Econometrica 41, 587–601 (1973)
9. Kelly, J.S.: Arrow Impossibility Theorems. Academic, New York (1978)
10. Malawski, M., Zhou, L.: A note on social choice theory without the Pareto principle. Soc. Choice Welf. **11**, 103–107 (1994)
11. McMorris, F.R., Neumann, D.: Consensus functions defined on trees. Math. Soc. Sci. **4**, 131–136 (1983)
12. McMorris, F.R., Powers, R.C.: Consensus functions on tree quasi-orders that satisfy an independence condition. Math. Soc. Sci. **48**, 183–192 (2004)
13. Mirkin, B.: Group Choice. Scripta Series in Mathematics. V.H. Winston, Washington (1979)
14. Muller, E., Satterthwaite, M.A.: The equivalence of strong positive association and strategy-proofness. J. Econ. Theory **14**, 412–418 (1977)

15. Reny, P.J.: Arrow's theorem and the Gibbard-Satterthwaite theorem: a unified approach. Econ. Lett. **70**, 99–105 (2001)
16. Satterthwaite, M.A.: Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. J. Econ. Theory **10**, 198–217 (1975)
17. Sholomov, L.A.: Explicit form of neutral social decision rules for basic rationality conditions. Math. Soc. Sci. **39**, 81–107 (2000)
18. Wilson, R.: Social choice theory without the Pareto principle. J. Econ. Theory **5**, 478–486 (1972)

Weak Hierarchies: A Central Clustering Structure

Patrice Bertrand and Jean Diatta

Abstract The k-weak hierarchies, for $k \geq 2$, are the cluster collections such that the intersection of any $(k+1)$ members equals the intersection of some k of them. Any cluster collection turns out to be a k-weak hierarchy for some integer k. Weak hierarchies play a central role in cluster analysis in several aspects: they are defined as the 2-weak hierarchies, so that they not only extend directly the well-known hierarchical structure, but they are also characterized by the rank of their closure operator which is at most 2. The main aim of this chapter is to present, in a unique framework, two distinct weak hierarchical clustering approaches. The first one is based on the idea that, since clusters must be isolated, it is natural to determine them as weak clusters defined by a positive weak isolation index. The second one determines the weak subdominant quasi-ultrametric of a given dissimilarity, and thus an optimal closed weak hierarchy by means of the bijection between quasi-ultrametrics and (indexed) closed weak hierarchies. Furthermore, we highlight the relationship between weak hierarchical clustering and formal concepts analysis, through which concept extents appear to be weak clusters of some multiway dissimilarity functions.

Keywords Weak hierarchy • Quasi-ultrametric • 2-Ball • Weak cluster • Formal concept

P. Bertrand (✉)
CEREMADE, Université Paris Dauphine, Paris, France
e-mail: bertrand@ceremade.dauphine.fr

J. Diatta
LIM-EA2525, Université de la Réunion, Saint-Denis, France
e-mail: jean.diatta@univ-reunion.fr

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_14,

1 Introduction

Cluster analysis, also named clustering, is a basic unsupervised learning task which is generally understood to be the search for groups in the data set in such a way that the obtained groups, called clusters, are both homogeneous and well separated. In a general setting, the degrees of homogeneity and separation of clusters are computed from a two-way map, called dissimilarity, that generalizes the usual notion of distance defined between any two entities of the data set. Partition and hierarchy of clusters,[1] also called simply hierarchy, are the most known types of structure in cluster analysis. Since any two clusters of a partition and of a hierarchy are either disjoint or nested, one major limitation of these structures is that they do not allow any overlap between two clusters, which is a drawback since real data sets may include overlapping clusters. Since the 1980s, several extensions of the set of hierarchies have been investigated in order to allow overlapping clusters, e.g. [2, 4, 9, 18, 20, 22, 23]. Among these extensions, two have been considered in several papers, namely the pyramids, also called sometimes pseudo-hierarchies, and the weak hierarchies. Given a data set E, a pyramid is a collection of subsets of E for which there exists a total order, defined on E, such that each cluster is an interval of this order. A weak hierarchy is any collection $\mathcal{W}$ of subsets of E such that $A \cap B \cap C \in \{A \cap B, B \cap C, A \cap C\}$ for all members A, B, C of $\mathcal{W}$. It is clear that pyramids are a particular case of weak hierarchies. Since a hierarchy can be defined as any collection $\mathcal{H}$ of subsets of E such that $A \cap B \in \{A, B, \emptyset\}$ for all members A, B of $\mathcal{H}$, weak hierarchies are a natural extension of the hierarchies. In addition, they admit interesting combinatorial properties [2, 4, 18] such as the property that the closure operator associated with a weak hierarchy has rank at most 2. The purpose of this chapter is to present clustering algorithms that generate a weak hierarchy defined on a data set E, provided that E is equipped with a given dissimilarity. One of these algorithms, is based on the idea that, since clusters must be homogeneous, it is natural to define clusters as the subsets that are convex in some abstract sense. A different approach, proposed in [13], consists in approximating the given dissimilarity by a quasi-ultrametric, since quasi-ultrametrics and (indexed) closed weak hierarchies are in one-one correspondence.

The content of this text is as follows: next section reminds to the reader some basic terminology used in the theory of clustering and elementary properties of weak hierarchies. Section 3 provides two different algorithms of weak hierarchical clustering, each of them being derived from a different definition of plausible clusters, i.e. subsets that are homogeneous and/or well separated according to an arbitrary given dissimilarity. Section 4 presents a different type of clustering

[1] The hierarchical structure is a highly versatile structure, as attested by hierarchies in cluster analysis, ontologies in knowledge representation, decision trees in supervised classification and by tree-based data structures such as PQ-trees.

algorithm [13] in the sense that it consists first in computing the weak subdominant quasi-ultrametric of a dissimilarity, and then in generating the unique closed weak hierarchy associated with this quasi-ultrametric approximation. Last section is devoted to a link between weak hierarchies and Galois lattices.

2 Background

2.1 Dissimilarities and Standard Subsets

In the following, E denotes the ground set which is assumed to be arbitrary and finite. A dissimilarity designates any map $d : E \times E \mapsto \mathbb{R}^+$ such that, for each x, y in $E \times E$, we have

$$d(x, y) = d(y, x) \geq d(x, x) = 0.$$

A dissimilarity is said to be *proper* if $d(x, y) = 0$ implies $x = y$. Let x, y be two (not necessarily distinct) elements of E, d be an arbitrary dissimilarity defined on E, and r be a nonnegative real number. The d-ball (or simply ball) of center x and radius r is the set $B^d(x, r)$ (or simply $B(x, r)$) of elements of E whose d-dissimilarity degree from x is at most r, i.e., formally,

$$B^d(x, r) = B(x, r) = \{z \in E : d(x, z) \leq r\}.$$

The $(d, 2)$-ball (or simply 2-ball) generated by x, y is the set denoted as B^d_{xy} (or simply B_{xy}) and defined by

$$B^d_{xy} = B_{xy} = B(x, d(x, y)) \cap B(y, d(x, y)).$$

Figure 1 illustrates these notions in the case of an Euclidean dissimilarity function.

The *diameter* of a nonempty subset A of E, denoted as $\mathrm{diam}_d\, A$, or $\mathrm{diam}\, A$ if there is no ambiguity on the choice of d, is defined as $\mathrm{diam}\, A = \max\{d(a, b) : a, b \in A\}$. A subset M of E is said to be *maximally linked in the sense of* d, or shortly an M_L*-set*, if for all subset N such that $M \subset N$, we have $\mathrm{diam}_d\, M <$

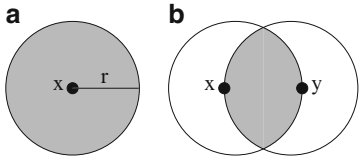


Fig. 1 (**a**) A ball of center x and radius r and (**b**) a 2-ball generated by x, y

$\mathrm{diam}_d\, N$. A subset M is said to be *maximally linked at level* h (in the sense of d) if it is maximally linked and if $\mathrm{diam}_d\, M = h$. It is easily checked that a subset of E is maximally linked if and only if it is a maximal clique for at least one threshold graph associated with a symmetric binary relation $Td(h)$ (with $h \in \mathbb{R}^+$) that is defined by $Td(h) = \{(a,b) : d(a,b) \leq h\}$. The collection of maximally linked (in the sense of d) subsets is denoted as $M_L(Td)$. There exists a close relationship between the M_L-sets and the 2-balls. For any $x, y \in E$, let $\mathcal{M}(x,y)$ denote the set defined by $\mathcal{M}(x,y) = \{M \in M_L(Td) : x, y \in M, \mathrm{diam}\, M = d(x,y)\}$. Then it can be proved [10] that we have

$$B_{xy} = \bigcup \mathcal{M}(x,y).$$

Consider now two elements of E and let M be an M_L-set of level h. Because the dissimilarity degree between these two elements is either less than h if both of them belong to M, or greater than h if only one of them belongs to M, maximally linked subsets are both homogeneous and well separated, and thus they are good candidate for being clusters. Furthermore, there exists a fundamental bijection between collections of M_L-sets and dissimilarities. More precisely, the general correspondence Φ that associates each dissimilarity d with the pair $(M_L(Td), \mathrm{diam}_d)$ is a bijection between the set of dissimilarities on E and the collection of pairs of the form $(\mathcal{F}, f)$ where $\mathcal{F}$ denotes any collection of nonempty subsets of E that contains E and where $f : \mathcal{F} \mapsto \mathbb{R}^+$ is an increasing map from $(\mathcal{F}, \subseteq)$ to $(\mathbb{R}^+, \leq)$ satisfying the so-called Gilmore condition and another technical condition (see [5, 7] for more details). The bijection Φ is, indeed, an extension of bijections that were established between various classes of dissimilarities and various types of clustering structures: we will give examples of such bijections in Sects. 2.2 and 2.3.

In what follows, we will consider two types of dissimilarities. First, we will consider the well-known ultrametrics: a dissimilarity on E is called an *ultrametric* if for all $x, y, z \in E$,

$$d(x,z) \leq \max\{d(x,y), d(y,z)\}. \tag{U}$$

Ultrametrics are a particular type of quasi-ultrametrics. A *quasi-ultrametric* is any dissimilarity on E which satisfies the so-called four points inequality [1], i.e. such that, for all $x, y, z, t \in E$,

$$\max\{d(z,x), d(z,y)\} \leq d(x,y) \;\Rightarrow\; d(z,t) \leq \max\{d(t,x), d(t,y), d(x,y)\}. \tag{QU}$$

It is well known that the ultrametric inequality (U) admits a simple geometric interpretation: (U) is equivalent to assert that each triangle is isosceles with the length of the basis less than or equal to the common length of the two other sides.

The four points inequality (QU) is equivalent to the conjunction of two conditions: the diameter condition and the inclusion condition [18]. These two conditions are defined as follows:

- Inclusion condition: $\forall a, b \in E, B_{xy} \subseteq B_{ab}$, for all $x, y \in B_{ab}$;
- Diameter condition: $\forall a, b \in E, \text{diam}\, B_{ab} = d(a, b)$.

2.2 Hierarchies

A collection $\mathcal{H}$ of subsets of a finite entity set E is said to be a *strong hierarchy* if its members are pairwise either disjoint or nested, i.e.:

(H1) Two members X, Y of $\mathcal{H}$ are either disjoint or nested, or equivalently, $X \cap Y \in \{\varnothing, X, Y\}$.

The well-known *hierarchies* are a type of strong hierarchies. More precisely, a *hierarchy*, also called hierarchy of clusters, designates any strong hierarchy which satisfies the following two conditions:

(H2) $E \in \mathcal{H}$ and $\varnothing \notin \mathcal{H}$;
(H3) The minimal members of $\mathcal{H}$ partition E.

Each hierarchy is usually visualized by its associated *Hasse diagram*, which is a kind of tree diagram that represents the covering relation of the set inclusion order defined on the set of clusters.

An *indexed hierarchy* designates any pair $(\mathcal{H}, f)$ where $\mathcal{H}$ is a hierarchy and $f : \mathcal{H} \mapsto \mathbb{R}^+$ is an increasing map, i.e. $f(A) < f(B)$ whenever the strict inclusion $A \subset B$ holds true for any two clusters A, B. From a practical point of view, the index f is used to indicate the degree of heterogeneity of each cluster. An indexed hierarchy $(\mathcal{H}, f)$ is represented by a weighted Hasse diagram, called *dendrogram*, that is a Hasse diagram of $\mathcal{H}$ where each node A is displayed at a height (in the tree diagram) which is proportional to its weight $f(A)$. If some partition of the data set exists such that clusters are well separated with respect to the dissimilarity related to the data set E, then most hierarchical clustering methods provide dendrograms that enable to detect graphically this partition of the data set.

Let us recall the well-known bijection between ultrametrics and indexed hierarchies. Note that in the case where a dissimilarity d is ultrametric, then the M_L-sets of d coincide with the balls of d.

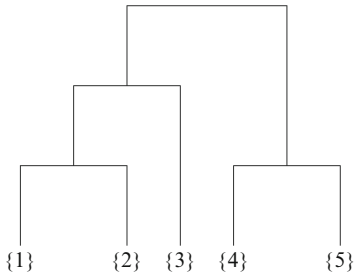
Proposition 1 ([6, 25]). *The restriction of Φ to the set $\mathcal{U}$ of ultrametrics defines a bijection from $\mathcal{U}$ onto the set of indexed hierarchies.*

By condition (H1), the members of hierarchies don't overlap. This makes them lack to represent situations where two properly intersecting entity subsets share features, as can be observed in Table 1 which presents a data set, say $\mathcal{D}$, about five market baskets and five items: bread (brd), butter (btr), cheese (chs), eggs (egg), milk (mlk);

Table 1 Example of data set: description of five basket entities

	brd	btr	chs	egg	mlk
1		x	x		x
2	x	x	x		x
3		x	x		
4		x	x	x	
5	x	x	x	x	

Fig. 2 Hasse diagram of hierarchy $\mathcal{H}_1$



for instance, the market basket labeled 1 contains butter, cheese and milk. Basket sets $X_1 := \{1, 2, 3\}$, $X_2 := \{3\}$ and $X_3 := \{3, 4, 5\}$ can never be members of the same hierarchy despite the fact that items characterizing basket 3 (butter and cheese) are shared by baskets in X_1 and X_3.

Figure 2 represents a hierarchy $\mathcal{H}_1$ on the 7-element set $E_1 := \{1, 2, 3, 4, 5, 6, 7\}$. The leaves (bottom-most level) are minimal members of $\mathcal{H}_1$: $\{1\}, \{2\}, \{3\}, \ldots$; every internal node is the union of its sons.

2.3 Weak Hierarchies

Weak hierarchies have been independently introduced, in the framework of cluster analysis, by Batbedat, under the name "Médinclus hypergraps," and by Bandelt and Dress [2], in the fall 1980s. Bandelt and Dress called them weak hierarchies since they are defined by weakening the nestedness condition (H1) that characterizes the so-called strong hierarchies. A *weak hierarchy* on E is a collection $\mathcal{W}$ of subsets of E, satisfying:

(WH) The intersection of any three members X, Y, Z of $\mathcal{W}$ is always the intersection of two of these three, i.e., $X \cap Y \cap Z \in \{X \cap Y, X \cap Z, Y \cap Z\}$.

Condition (WH) is equivalent to the following forbidden configuration: there are no three members X_1, X_2, X_3 and three elements x_1, x_2, x_3 such that $x_i \in X_j$ if and only if $i \neq j$. Figure 3 represents a weak hierarchy $\mathcal{W}_2$ on the 4-element set $E_2 := \{1, 2, 3, 4\}$.

A weak hierarchy $\mathcal{W}$ is said to be *closed* if it is closed under nonempty finite intersections, in other words if, for all $A, B \in \mathcal{W}$, we have $A \cap B \in \mathcal{W} \cup \{\emptyset\}$. Given

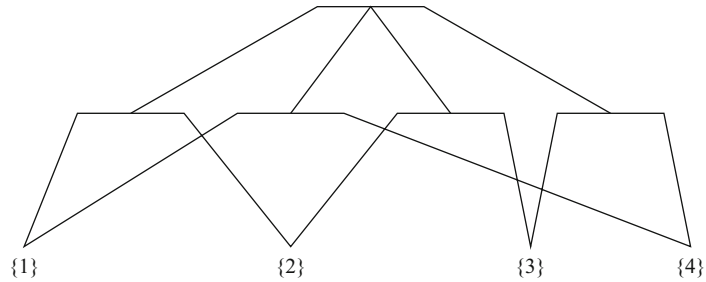


Fig. 3 A weak hierarchy

an arbitrary weak hierarchy $\mathcal{W}$, its closure under nonempty finite intersections will be denoted as $\hat{\mathcal{W}}$. Given an arbitrary collection $\mathcal{F}$ of subsets of E, we will denote as $\langle \quad \rangle_{\mathcal{F}}$ the closure operator that associates each subset A of E with $\bigcap\{F \in \mathcal{F} : A \subseteq F\}$. With these notations, we may then formulate two main combinatorial properties of weak hierarchies that emphasize the central role of the weak hierarchical structure in the theory of clustering.

Proposition 2 ([2]). *Given a collection $\mathcal{F}$ of subsets of E, the following conditions are equivalent:*

(i) *$\mathcal{F}$ is a weak hierarchy;*
(ii) *The operator $\langle \ \rangle_{\mathcal{F}}$ has rank at most 2, i.e. for each nonempty subset A of E, there exist $a, b \in A$ such that $\langle A \rangle_{\mathcal{F}} = \langle a, b \rangle_{\mathcal{F}}$;*
(iii) *$\hat{\mathcal{F}}$ is a weak hierarchy.*

Denoting as $B_2(d)$ the set of 2-balls, in the sense of a dissimilarity d, the following property holds true.

Proposition 3 ([7]). *Given a proper dissimilarity d, the following conditions are equivalent:*

(i) *$M_L(Td) = B_2(d)$*
(ii) *The collection $B_2(d)$ is closed;*
(iii) *$M_L(Td)$ is a closed weak hierarchy.*

We close this section with the result of a one-to-one correspondence between the class of quasi-ultrametrics and the indexed closed weak hierarchies.

Proposition 4 ([18]). *The restriction of Φ to the set $\mathcal{Q}$ of quasi-ultrametrics defines a bijection from $\mathcal{Q}$ onto the set of indexed closed weak hierarchies.*

3 Obtaining a Weak Hierarchy from a Dissimilarity Measure

3.1 Weak Clusters

Dissimilarity functions play an important role in cluster analysis where they are often used for constructing clusters having a weak within-cluster and/or a strong between-cluster dissimilarity degrees [26]. Weak clusters introduced in [2] in the framework of pairwise similarity measures are among these clusters. They are said to be weak in contrast to the so-called strong clusters. A subset X of E is said to be a *strong* cluster associated with a pairwise dissimilarity function d (or d*-strong* cluster), if its d*-strong isolation index*

$$\mathbf{i}_d^s(X) := \min_{\substack{x,y\in X \\ z\notin X}} \{d(x,z) - d(x,y)\}$$

is strictly positive.

Figure 4 illustrates the configuration satisfied by a strong cluster associated with a pairwise dissimilarity function, say d: for all x, y within the cluster and z outside, each of the dissimilarities $d(x,z)$ and $d(y,z)$ is greater than the dissimilarity $d(x,y)$.

A nonempty subset X of E is said to be a *weak* cluster associated with a pairwise dissimilarity function d (or d*-weak* cluster), if its d*-weak isolation index*

$$\mathbf{i}_d^w(X) := \min_{\substack{x,y\in X \\ z\notin X}} \{\max\{d(x,z), d(y,z)\} - d(x,y)\}$$

is strictly positive. Figure 5 presents the configuration satisfied by a weak cluster associated with a pairwise dissimilarity function, say d: for all x, y within the cluster and z outside, at least one of the dissimilarities $d(x,z)$ and $d(y,z)$ is greater than the dissimilarity $d(x,y)$.

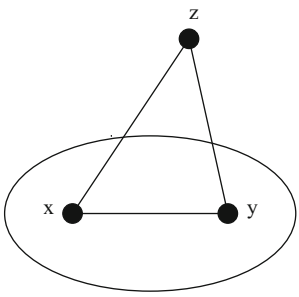


Fig. 4 Strong cluster associated with a pairwise dissimilarity measure

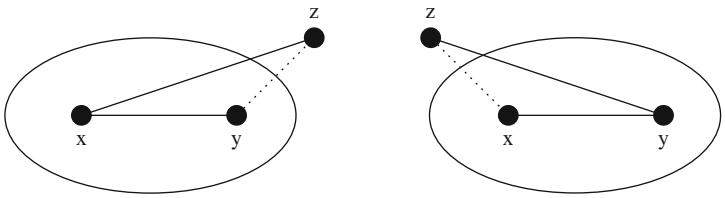


Fig. 5 Weak cluster associated with a pairwise dissimilarity measure

It should be noticed that any d-strong cluster is a d-weak one. Moreover, it is easily shown that the strong (resp. weak) clusters associated with a pairwise dissimilarity function form a strong (resp. weak) hierarchy [2].

Proposition 5. *Let d be a pairwise dissimilarity function on E. Then*

(i) *The strong clusters associated with d form a strong hierarchy called the* strong hierarchy associated with d.
(ii) *The weak clusters associated with d form a weak hierarchy called the* weak hierarchy associated with d.

3.2 Weak Clusters and 2-Balls

In Sect. 3.1, we have seen that weak hierarchies are related to dissimilarity functions via weak clusters. These weak clusters turn out to be special 2-balls. Below is a characterization of a weak cluster as a subset containing the 2-balls generated by each of the pairs of its (not necessarily distinct) elements [17].

Proposition 6 ([17]). *Let d be a dissimilarity function on E. A subset X of E is a d-weak cluster if and only if it satisfies the so-called* inclusion property*, i.e.,*

$$\forall x, y \in X : B^d_{xy} \subseteq X.$$

From Proposition 6, it can be easily derived that every nonempty weak cluster is a 2-ball.

Proposition 7 ([17]). *Let d be a dissimilarity function on E. If a subset X of E is a d-weak cluster, then $X = B^d_{xy}$, where x, y are such that $d(x, y) = \max_{u,v \in X} d(u, v)$.*

Finally, it follows from Propositions 5 and 7 that the weak hierarchy associated with a dissimilarity function d is the set of 2-balls of d satisfying the inclusion property, augmented with the empty set. This provides us with a way for specifying a weak hierarchy from any dissimilarity function. Moreover, it has been shown in [17] that nonempty members of any weak hierarchy are the 2-balls of some dissimilarity function.

Algorithm 1 WH(E, d)

Input: A finite nonempty entity set E and a dissimilarity measure d on E.
Output: The weak hierarchy $\mathcal{W}_d$ associated with d.

1: Set $\mathcal{W}_d := \varnothing$
2: **for** $i, j \in E$ **do**
3: **if** B_{ij}^d is not already considered **then**
4: LWI2B(i, j, E)
5: **if** B_{ij}^d is weakly isolated **then**
6: $\mathcal{W}_d \leftarrow \mathcal{W}_d \cup \{B_{ij}^d\}$
7: DWI2B$(i, j, \mathcal{W}_d)$
8: **end if**
9: **end if**
10: **end for**

3.3 Algorithms

3.3.1 The Bandelt and Dress Algorithm

Given a dissimilarity measure d on E, the following algorithm, which was proposed by Bandelt and Dress [2], computes the weak hierarchy associated with d. This algorithm can be described as follows. Assume that $E = \{e_1, \ldots, e_n\}$, one successively computes the weak hierarchy $\mathcal{W}_k^d$ associated with the restriction of d to $\{e_1, \ldots, e_k\}$ $(k \leq n)$. Put $\mathcal{W}_0^d = \varnothing$. If $\mathcal{W}_k$ is determined for $k < n$, then for each cluster C belonging to $\mathcal{W}_k$, one checks whether:

(a) $\max\{d(e_i, e_{k+1}), d(e_j, e_{k+1})\} > d(e_i, e_j)$ for all $i, j \leq k$ with $e_i, e_j \in C$,
(b) $\max\{d(e_i, e_j), d(e_j, e_{k+1})\} > d(e_i, e_{k+1})$ for all $i, j \leq k$ with $e_i \in C$ and $e_j \notin C$,

holds. Then, $\mathcal{W}_{k+1}^d$ contains C if and only if (a) holds, and it contains $C \cup \{e_{k+1}\}$ if and only if (b) holds. Finally, $\mathcal{W}_n^d$ is the weak hierarchy associated with d.

Bandelt and Dress observe that this algorithm executes in $O(n^5)$ even though it looks exponential.

3.3.2 A 2-Ball Convexity-Based Algorithm

Given a dissimilarity measure d on E, Algorithm 1 is designed for computing the set of 2-balls of d that satisfy the inclusion property, hence the weak hierarchy associated with d. The computation of a not already considered 2-ball B_{ij}^d depends on the fact that the pair $\{i, j\}$ is picked from a 2-ball B_{uv}^d already known to be weakly isolated or not. If $\{i, j\}$ is not picked from a 2-ball B_{uv}^d already known to be weakly isolated, Algorithm 2 is used for computing B_{ij}^d, supplying successively entities picked from E (line 4). Once a weakly isolated 2-ball B_{ij}^d is computed and B_{ij}^d inserted in the current set $\mathcal{W}_d$, Algorithm 3 is called on the triple $(i, j, \mathcal{W}_d)$ (lines 5–8).

Algorithm 2 LWI2B(i, j, X)

Input: A finite nonempty entity subset X and an entity pair $\{i, j\}$.
Output: The 2-ball B^d_{ij} if B^d_{ij} is weakly isolated.

1: Set $B^d_{ij} = \{i, j\}$
2: Set $\overline{B}^d_{ij} = \varnothing$
3: **for** $k \in X$ **do**
4: **if** $\max\{d(i,k), d(j,k)\} \leq d(i,j)$ **then**
5: $B^d_{ij} \leftarrow B^d_{ij} \cup \{k\}$
6: **for** $u \in B^d_{ij}$ and $v \in \overline{B}^d_{ij}$ **do**
7: **if** $\max\{d(k,v), d(u,v)\} \leq d(k,u)$ **then**
8: mark B^d_{ij} as already considered and go to STOP
9: **end if**
10: **end for**
11: **else**
12: $\overline{B}^d_{ij} \leftarrow \overline{B}^d_{ij} \cup \{k\}$
13: **for** $u, v \in B^d_{ij}$ **do**
14: **if** $\max\{d(u,k), d(v,k)\} \leq d(u,v)$ **then**
15: mark B^d_{ij} as already considered and go to STOP
16: **end if**
17: **end for**
18: **end if**
19: **end for**
20: mark B^d_{ij} as weakly isolated
21: mark B^d_{ij} as already considered
22: STOP

Algorithms 3 and 2 compute a weakly isolated 2-ball, say B^d_{ij}, depending on the entities i and j are chosen or not from a 2-ball known to be weakly isolated. Let us first describe Algorithm 2. It computes the 2-ball B^d_{ij} when B^d_{ij} is weakly isolated, supplying entities successively picked from an entity set X. The 2-ball B^d_{ij} is initially set to its generator (line 1), and its complement $\overline{B}^d_{ij}$ is initially set to the empty set (line 2). Then entities k are picked from X and tested to know whether they belong to B^d_{ij} or $\overline{B}^d_{ij}$ (line 4). After assigning k to either B^d_{ij} or $\overline{B}^d_{ij}$ (line 5 or 12), we test whether the current set B^d_{ij} is weakly isolated relatively to the current set $\overline{B}^d_{ij}$; if this test is negative, the construction of B^d_{ij} is stopped (lines 6–10 or 13–17). If $B^d_{ij} \cup \overline{B}^d_{ij} = X$, then B^d_{ij} is weakly isolated. The following straightforward property will be useful in the computation of weakly isolated 2-balls.

Proposition 8. *Let d be a dissimilarity function on E. Let B^d_{xy} be a weakly isolated 2-ball containing u, v. If $d(u,v) \geq d(x,y)$ and B^d_{uv} is weakly isolated, then $B^d_{uv} = B^d_{xy}$.*

Algorithm 3 DWI2B$(i, j, \mathcal{F})$

Input: An entity pair $\{i, j\}$ and a (potentially empty) set $\mathcal{F}$ of 2-balls.
Output: A (potentially empty) set $\mathcal{F}$ of 2-balls B^d_{uv}, where $u, v \in B^d_{ij}$ and B^d_{uv} is weakly isolated.

```
1: for u, v ∈ B^d_ij do
2:   if B^d_uv is not already considered then
3:     if d(i, j) ≤ d(u, v) then
4:       mark B^d_uv as already considered
5:     else
6:       LWI2B(u, v, B^d_ij)
7:       if B^d_uv is weakly isolated then
8:         F ← F ∪ {B^d_uv}
9:         DWI2B(u, v, F)
10:      end if
11:    end if
12:  end if
13: end for
```

Table 2 A dissimilarity data on a six-element entity set

x	0					
y	3	0				
z	4	4	0			
t	5	2	3	0		
u	1	2	5	1	0	
v	4	1	5	6	1	0
	x	y	z	t	u	v

In Algorithm 3, B^d_{ij} is assumed to be weakly isolated. For $u, v \in B^d_{ij}$, either (1) $d(i, j) \leq d(u, v)$ or (2) $d(i, j) > d(u, v)$. In case (1), if B^d_{uv} was weakly isolated, then by Proposition 8, we would have $B^d_{uv} = B^d_{ij}$ and there would be no need to keep B^d_{uv} since B^d_{ij} is assumed to be already considered (lines 3–4). In case (2), Algorithm 2 is used for the computation of B^d_{uv} (if weakly isolated), supplying entities picked successively from B^d_{ij} (line 6). Indeed, as B^d_{ij} is assumed to be weakly isolated, $B^d_{uv} \subseteq B^d_{ij}$ by Proposition 6. Therefore, entities outside B^d_{ij}, hence outside B^d_{uv}, cannot prevent B^d_{uv} from being weakly isolated. If B^d_{uv} is weakly isolated, then Algorithm 2 is recursively called on the triple $(u, v, \mathcal{F})$ after insertion of the 2-ball B^d_{uv} in $\mathcal{F}$ (lines 7–10).

Example 1. As an example, let us consider the following dissimilarity measure, say d, on the 6-element entity set $E = \{x, y, z, t, u, v\}$, which is defined in Table 2. The weak hierarchy associated with d is given in Table 3.

Table 3 The weak hierarchy associated with the dissimilarity given in Table 2

Cluster size	Clusters
0	$\emptyset$
1	$\{x\}$; $\{y\}$; $\{z\}$; $\{t\}$; $\{u\}$; $\{v\}$
2	$\{x, u\}$; $\{y, v\}$; $\{z, t\}$; $\{t, u\}$; $\{u, v\}$
6	$\{x, y, z, t, u, v\}$

4 Obtaining a Weak Hierarchy via the Weak Subdominant Quasi-Ultrametric of a Dissimilarity

As previously seen in Sect. 2, there exists a bijection between quasi-ultrametrics and indexed closed weak hierarchies. This bijection has been studied in detail in [17]. It associates each quasi-ultrametric, say d, with the indexed closed weak hierarchy $(\mathcal{W}, f) = (B_2(d), \mathrm{diam}_d)$. Conversely, the inverse bijection associates each indexed closed weak hierarchy $(\mathcal{W}, f)$ with its induced dissimilarity which is here a quasi-ultrametric. Therefore, an approach to achieve a weak hierarchical clustering consists in: first, determine a quasi-ultrametric which is an approximation of the dissimilarity given as an input data, and then compute the indexed weak hierarchy associated with the obtained quasi-ultrametric. This section is devoted to an approach of this type, proposed by Brucker [13], which aims to determine a weak subdominant quasi-ultrametric of any dissimilarity d, the weak subdominant being an extension of the more classical notion of subdominant.

4.1 Subdominant, Weak Subdominant of a Dissimilarity

Let us consider the set $\mathcal{D}$ of all dissimilarities on E. The set $\mathcal{D}$ is endowed with the point-wise order denoted as $\preceq$ and defined as follows. Given two dissimilarities d_1 and d_2 defined on E, it is said that $d_1 \preceq d_2$ if for all $x, y \in E$, we have $d_1(x, y) \leq d_2(x, y)$. The binary relation $\preceq$ is clearly a partial order defined on $\mathcal{D}$. Given a subset $\mathcal{D}'$ of $\mathcal{D}$ and an arbitrary dissimilarity $d \in \mathcal{D}$, we consider the down set $(\downarrow d)$ defined by

$$\downarrow d = \{d' \in \mathcal{D}' \ : \ d' \preceq d\}.$$

The set $\mathcal{U}$ of ultrametrics on E and the set $\mathcal{Q}$ of quasi-ultrametrics on E will be considered later as examples of subset $\mathcal{D}'$. From a general point of view, several types of lower $\mathcal{D}'$-approximations can be investigated according to the three following possible disjoint cases:

1. $(\downarrow d)$ admits a (unique) greatest element. In this case, this greatest element is, by definition, called the *subdominant in* $\mathcal{D}'$ of d, or simply the *subdominant* of d if there is no ambiguity on the choice of $\mathcal{D}'$. It is well known that each dissimilarity d admits a subdominant in $\mathcal{U}$, which is called the *subdominant ultrametric* of d.

Table 4 The weak subdominant quasi-ultrametric $q(d)$ of the dissimilarity d given in Table 2

x	0					
y	3	0				
z	4	4	0			
t	3	2	3	0		
u	1	2	4	1	0	
v	3	1	4	2	1	0
	x	y	z	t	u	v

2. $(\downarrow d)$ admits only one maximal element, but this maximal element is not the greatest. Then this unique maximal element will be called the *weak subdominant* in $\mathcal{D}'$ of d, or simply the *weak subdominant* of d if there is no ambiguity on the choice of $\mathcal{D}'$. Such a case happens when $(\downarrow d)$ not only admits a maximal dissimilarity, say $d^\star$, but also contains dissimilarities that are incomparable with $d^\star$ and whose set is not bounded w.r.t. order $\preceq$. It was proved in [13] that each dissimilarity d admits a weak subdominant in $\mathcal{Q}$, which is called the *weak subdominant quasi-ultrametric* of d; see next Sect. 4.2 for more details.
3. $(\downarrow d)$ admits more than one maximal elements. In this case, such maximal dissimilarities can be said to be *lower maximal dissimilarities* in $\mathcal{D}'$ w.r.t. d, or simply the *lower maximal dissimilarities* w.r.t. d if there is no ambiguity on the choice of $\mathcal{D}'$. It was proved in [8] that each dissimilarity d admits lower maximal dissimilarities in the set $\mathcal{D}'$ of the so-called paired-ultrametrics.

4.2 Algorithm for Computing the Weak Subdominant Quasi-Ultrametric

We now present Algorithm 4 which is proposed in [13]. This algorithm provides a constructive proof of the existence of the weak subdominant quasi-ultrametric of any dissimilarity.

Let us call *quatuor* of E any subset Q of E of size 4. Moreover, given any $x, y \in E$, let us denote as $\mathbb{Q}[x, y]$ the set of all quatuors containing both x and y. A quatuor Q will be said to be quasi-ultrametric for d whenever the restriction of d to Q is quasi-ultrametric. In order to introduce Algorithm 4, we need two more notations. For any subset A of size at least 2, we denote as $A^{(2)}$ the set of pairs of (distinct) elements in A. Note that if Q is a quatuor, then $|\, Q^{(2)} \,| = 6$. We also denote $n = |\, E \,|$.

Note that, in step 8 of Algorithm 4, the set $\arg\max\{d_i(u, v) \ : \ u, v \in Q\}$ is reduced to a single element since, if Q is a non-quasi-ultrametrical quatuor for d_i, then there exists a unique pair (u_Q, v_Q) from Q such that $d_i(u_Q, v_Q) = \text{diam} d_i(Q)$ (see Proposition 6(2) in [13]).

It is easily checked that the complexity of this algorithm is $\mathcal{O}(n^4)$. The next technical lemma is useful in order to prove the main result provided by

Algorithm 4 WSDQ(E, d)

Input: A finite set E and a dissimilarity d defined on E.
Output: The weak subdominant quasi-ultrametric of d.

```
Set F_0 := ∅
Set d_0 := d
for i = 0 to n(n − 1)/2 do
    Set (x, y) ∈ arg min{d_i(u, v) : u, v ∉ F_i}
    d_{i+1} = d_i
    for Q ∈ ℚ[x, y] do
        if Q is not quasi-ultrametrical for d_i AND |Q^(2) ∩ F_i| = 4 then
            Set (u_Q, v_Q) = arg max{d_i(u, v) : u, v ∈ Q}
            d_{i+1}(u_Q, v_Q) = d_i(x, y)
        end if
    end for
    F_{i+1} = F_i ∪ {{x, y}}
end for
return d_{n(n−1)/2}
```

Proposition 9. Note that property (i) is trivial and that (ii) implies that the restriction of d_i to subset F_i is a quasi-ultrametric.

Lemma 1 ([13]). *Denote $m = n(n-1)/2$ and $[m] = \{0, 1, \ldots, n(n-1)/2\}$. Then, using the notations defined in Algorithm 4, the following properties hold true.*

(i) *If $i < m$, then $F_i \subsetneq F_{i+1}$ and $d_{i+1} \leq d_i$, so that $F_m = E^{(2)}$.*
(ii) *If $(x, y) = \arg\min\{d_i(u, v) : u, v \notin F_i\}$ and if $Q \in \mathbb{Q}[x, y]$ and $|Q^{(2)} \cap F_i| \geq 5$, then Q is quasi-ultrametrical for d_i.*
(iii) *Let $d' \leq d$ be quasi-ultrametric. If there exists $i \in [m]$ and $\{x, y\} \in E^{(2)}$ such that $d'(x, y) > d_i(x, y)$, then there exists $\{u, v\} \in E^{(2)}$ such that $d'(u, v) < d_j(u, v)$ for all $j \in [m]$.*

As a nontrivial consequence of Lemma 1, one can establish the following result which proves that Algorithm 4 constructs the weak subdominant quasi-ultrametric of any dissimilarity.

Proposition 9 ([13]). *Let d be an arbitrary dissimilarity defined on E. Then $d_{n(n-1)/2}$ is the weak subdominant quasi-ultrametric of d.*

Example 2 (Continuation). Let us consider again the dissimilarity d whose values are displayed in Example 1 (Sect. 3.3.2). Applying Algorithm 4, we then obtain the weak subdominant quasi-ultrametric of d (Table 4).

The closed weak hierarchy associated with $q(d)$ is given in Table 5: its clusters are exactly the 2-balls of $q(d)$.

In this example, one can notice that the closed weak hierarchy associated with $q(d)$ (cf. Table 5) does not coincide with the closed weak hierarchy associated with d (cf. Table 3). However, there exist also dissimilarities d such that these closed weak hierarchies, which are associated, respectively, with d and $q(d)$, coincide (e.g., consider the dissimilarity d provided in Table 1 in [13]).

Table 5 The (closed) weak hierarchy associated with dissimilarity $q(d)$ given in Table 4

Cluster size	Clusters
0	$\emptyset$
1	$\{x\}$; $\{y\}$; $\{z\}$; $\{t\}$; $\{u\}$; $\{v\}$
2	$\{x, u\}$; $\{y, v\}$; $\{z, t\}$; $\{u, v\}$; $\{t, u\}$
4	$\{y, t, u, v\}$
5	$\{x, y, t, u, v\}$
6	$\{x, y, z, t, u, v\}$

5 Links to Formal Concept Analysis

5.1 Galois Lattices

5.1.1 The Galois Lattice of a Binary Context

A binary relation from a set E to a set F is a triple (E, F, R), where R is a subset of the cross product $E \times F$. In formal concept analysis, a so-called formal context is a binary relation (E, F, R), where elements of E are called objects and those of F attributes [28]. Thus, formal contexts are sometimes called *binary contexts*.

Let $\mathbb{K} := (E, F, R)$ be a binary context. Then $\mathbb{K}$ induces a Galois correspondence between the partially ordered sets (posets) $(\mathcal{P}(E), \subseteq)$ and $(\mathcal{P}(F), \subseteq)$ by means of the maps

$$f : X \mapsto \bigcap_{x \in X} \{y \in F : (x, y) \in R\}$$

and

$$g : Y \mapsto \bigcap_{y \in Y} \{x \in E : (x, y) \in R\}.$$

For $X \subseteq E$, $f(X)$ is the set of attributes common to objects in X, and for $Y \subseteq F$, $g(Y)$ is the set of objects that share attributes in Y. The Galois correspondence (f, g) induces, in turn, a closure operator $\varphi := g \circ f$ on $(\mathcal{P}(E), \subseteq)$ [11]. That is:

(C1) $X \subseteq \varphi(X)$;
(C2) $X \subseteq Y$ implies $\varphi(X) \subseteq \varphi(Y)$;
(C3) $\varphi(\varphi(X)) = \varphi(X)$.

A pair $C := (X, Y) \in \mathcal{P}(E) \times \mathcal{P}(F)$ such that $\varphi(X) = X$ and $f(X) = Y$ is called a *formal concept* of $\mathbb{K}$. The entity set X is the extent of C and Y its

intent. The set $G(\mathbb{K})$ of formal concepts of $\mathbb{K}$, endowed with the order defined by $(X_1, Y_1) \leq (X_2, Y_2)$ if and only if $X_1 \subseteq X_2$ (or, equivalently $Y_2 \subseteq Y_1$), is a complete lattice called the *Galois lattice* of the binary context $\mathbb{K}$ [3] or the *concept lattice* of the formal context $\mathbb{K}$ [28].

Example 1. The data set given in Fig. 1 can be viewed as representing a binary context $\mathbb{K}_1 = (E_1, F_1, R_1)$, where, for instance, E_1 is the set of five market baskets, F_1 the set of five items, and where R_1 relates a market basket with an item if that item is contained in the basket in question. The pair $(\{1, 2\}, \{\text{btr}, \text{chs}, \text{mlk}\})$ belongs to the Galois lattice of $\mathbb{K}_1$; but $(\{2, 3\}, \{\text{btr}, \text{chs}\})$ does not belong to $G(\mathbb{K}_1)$ because $\{2, 3\}$ is not a fixed point of φ since the basket labeled 1 contains the items "egg" and "cheese" shared by baskets 2 and 3.

5.1.2 The Galois Lattice of a Meet-Closed Description Context

A meet-closed description context is a context where entities are described in a meet-semilattice. Meet-closed description contexts have been considered by several authors under various names [14, 19, 24]. We will denote such a context as a triple $(E, \mathcal{D}, \delta)$, where E is the entity set, $\mathcal{D}$ the entity description space, and δ a descriptor that maps E into $\mathcal{D}$. A meet-closed description context $\mathbb{K} := (E, \mathcal{D}, \delta)$ induces a Galois connection between $(\mathcal{P}(E), \subseteq)$ and $\mathcal{D}$ by means of the maps

$$f : X \mapsto \wedge \{\delta(x) \,:\, x \in X\}$$

and

$$g : \omega \mapsto \{x \in E \,:\, \omega \leq \delta(x)\},$$

for $X \subseteq E$ and $\omega \in \mathcal{D}$. Then, in these conditions, the map $\varphi_\delta := g \circ f$ is a closure operator on $\mathcal{P}(E)$. A subset X of E is said to be *φ_δ-closed* (or a *Galois closed entity set* (of $\mathbb{K}$) under φ_δ) when $\varphi_\delta(X) = X$. The Galois lattice of a meet-closed description context is defined in a similar way as that of a binary context.

Galois closed entity sets play an important role in classification because they provide easy-to-interpret clusters [21].

When $\mathcal{D}$ is a join-semilattice, the join-closed description context $(E, \mathcal{D}, \delta)$ induces a Galois connection between $(\mathcal{P}(E), \subseteq)$ and the order-dual of $\mathcal{D}$ by means of the maps

$$f^\partial : X \mapsto \vee \{\delta(x) \,:\, x \in X\}$$

and

$$g^\partial : \omega \mapsto \{x \in E \,:\, \omega \geq \delta(x)\},$$

for $X \subseteq E$ and $\omega \in \mathcal{D}$. Similarly, this Galois connection induces the closure operator $\varphi_\delta^\partial := g^\partial \circ f^\partial$ on $\mathcal{P}(E)$. Galois closed entity sets under φ_δ^∂ have been considered in the framework of symbolic data analysis [12, 27].

5.1.3 Galois Closed Entity Sets and Weak Clusters

In this section we present a result, established in [15], that links Galois closed entity sets to weak clusters associated with some pairwise or multiway dissimilarity measures. These dissimilarity measures satisfy a compatibility condition defined using a notion of valuation.

A *valuation* on a poset $(P, \leq)$ is a map $h : P \to \mathbb{R}_+$ such that $h(x) \leq h(y)$ when $x \leq y$. A *strict* valuation is a valuation h such that $x < y$ implies $h(x) < h(y)$. It may be noticed that an index f on a cluster structure $\mathcal{C}$ is nothing else than a strict valuation on the poset $(\mathcal{C}, \subseteq)$.

In all what follows, E will denote a finite entity set, $\mathcal{D}$ a meet-semilattice, δ a descriptor that maps E into $\mathcal{D}$, and $\mathbb{K}$ the meet-closed description context $(E, \mathcal{D}, \delta)$. For any $X \subseteq E$, $\delta(X)$ will denote the set of descriptions of entities belonging to X, and for any $x \in E$, $X + x$ will denote $X \cup \{x\}$. Let $\mathcal{I}_k(\delta(E))$ be the subset of $\mathcal{D}$ defined by

$$\mathcal{I}_k(\delta(E)) = \{\delta(x_1) \wedge \cdots \wedge \delta(x_k) : x_1, \ldots, x_k \in E\}.$$

Consider the map h_k^c defined on $\mathcal{I}_k(\delta(E))$ by

$$h_k^c(\omega) = \#\{\omega' \in \mathcal{I}_k(\delta(E)) \,:\, \omega' \leq \omega\},$$

i.e. the number of elements of $\mathcal{I}_k(\delta(E))$ which are less than or equal to ω. It is then easily observed that h_k^c is a strict valuation on $\mathcal{I}_k(\delta(E))$.

A pairwise dissimilarity measure d on E is said to be δ-meet compatible if there exists a valuation h on $\mathcal{I}_2(\delta(E))$ with which it is δ-meet compatible, i.e., such that $d(x, y) \leq d(u, v) \iff h(\delta(u) \wedge \delta(v)) \leq h(\delta(x) \wedge \delta(y))$.

If h is a strict valuation, d will be said to be *strictly* δ-meet compatible. The reader may observe that when $\mathcal{D}$ is a join-semilattice, a dual compatibility condition, say δ-join-compatibility, can be defined by reversing the right-hand side inequality in the above equivalence and replacing meets by joins.

Description-meet compatibility is a kind of natural agreement expressing the following fact: the less the descriptions of entities x and y have in common, the larger the dissimilarity degree between x and y.

To fix the ideas, assume that a part of entity description space is that depicted in Fig. 6. Then any δ-meet-compatible dissimilarity function d must satisfy the following inequalities: $d(x, u) \leq d(y, u) = d(x, y) \leq d(u, v) = d(x, v)$, $d(y, v) \leq d(u, v), \ldots$.

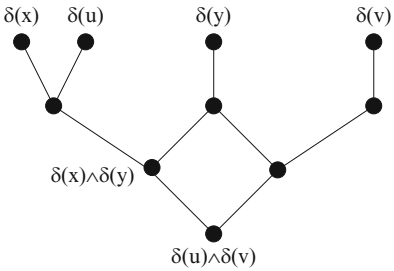


Fig. 6 A part of entity description space

The *canonical* δ-meet-compatible pairwise dissimilarity function on E is the dissimilarity $d^{h_2^c, M_2^c}$ defined by

$$d^{h_2^c, M_2^c}(x, y) = M_2^c - h_2^c(\delta(x) \wedge \delta(y)),$$

where $M_2^c = \max_{x \in E} h_2^c(\delta(x))$.

This notion of δ-meet compatibility generalizes naturally to multiway dissimilarity measures as well; see [16] for a detailed study of compatible multiway dissimilarities. Moreover, the following result shows that Galois closed entity sets are weak clusters.

Theorem 1 ([15]). *There is an integer $k \geq 2$ such that nonempty φ_δ-closed entity sets coincide with weak clusters associated with any strictly δ-meet compatible k-way dissimilarity measure.*

References

1. Bandelt, H.-J.: Four point characterization of the dissimilarity functions obtained from indexed closed weak hierarchies. In: Mathematisches Seminar. Universität Hamburg, Germany (1992)
2. Bandelt, H.-J., Dress, A.W.M.: Weak hierarchies associated with similarity measures: an additive clustering technique. Bull. Math. Biol. **51**, 113–166 (1989)
3. Barbut, M., Monjardet, B.: Ordre et classification. Hachette, Paris (1970)
4. Batbedat, A.: Les dissimilarités médas ou arbas. Statistique et Analyse des Données **14**, 1–18 (1988)
5. Batbedat, A.: Les isomorphismes HTS et HTE (après la bijection de Benzécri/Johnson) (première partie). Metron **46**, 47–59 (1988)
6. Benzécri, J.-P.: L'Analyse des données: la Taxinomie. Dunod, Paris (1973)
7. Bertrand, P.: Set systems and dissimilarities. Eur. J. Comb. **21**, 727–743 (2000)
8. Bertrand, P., Brucker, F.: On lower-maximal paired-ultrametrics. In: Brito, P., Bertrand, P., Cucumel, G., Carvalho, F.D. (eds.) Selected Contributions in Data Analysis and Classification, pp. 455–464. Springer, Berlin (2007)
9. Bertrand, P., Diday, E.: A visual representation of the compatibility between an order and a dissimilarity index: the pyramids. Comput. Stat. Q. **2**, 31–42 (1985)
10. Bertrand, P., Janowitz, M.F.: Pyramids and weak hierarchies in the ordinal model for clustering. Discrete Appl. Math. **122**, 55–81 (2002)

11. Birkhoff, G.: Lattice Theory. Colloquium Publications, vol. XXV, 3rd edn. American Mathematical Society, Providence (1967)
12. Brito, P.: Order structure of symbolic assertion objects. IEEE Trans. Knowl. Data Eng. **6**(5), 830–835 (1994)
13. Brucker, F.: Sub-dominant theory in numerical taxonomy. Discrete Appl. Math. **154**, 1085–1099 (2006)
14. Daniel-Vatonne, M.-C., Higuera, C.D.L.: Les termes: un modèle algébrique de repésentation et de structuration de données symboliques. Math. Inf. Sci. Hum. **122**, 41–63 (1993)
15. Diatta, J.: A relation between the theory of formal concepts and multiway clustering. Pattern Recognit. Lett. **25**, 1183–1189 (2004)
16. Diatta, J.: Description-meet compatible multiway dissimilarities. Discrete Appl. Math. **154**, 493–507 (2006)
17. Diatta, J., Fichet, B.: From Apresjan hierarchies and Bandelt-Dress weak hierarchies to quasi-hierarchies. In: Diday, E., Lechevalier, Y., Schader, M., Bertrand, P., Burtschy, B. (eds.) New Approaches in Classification and Data Analysis, pp. 111–118. Springer, Berlin (1994)
18. Diatta, J., Fichet, B.: Quasi-ultrametrics and their 2-ball hypergraphs. Discrete Math. **192**, 87–102 (1998)
19. Diatta, J., Ralambondrainy, H.: The conceptual weak hierarchy associated with a dissimilarity measure. Math. Soc. Sci. **44**, 301–319 (2002)
20. Diday, E.: Une représentation visuelle des classes empiétantes: les pyramides. Tech. Rep. 291, INRIA, France (1984)
21. Domenach, F., Leclerc, B.: On the roles of Galois connections in classification. In: Schwaiger, O.O.M. (ed.) Explanatory Data Analysis in Empirical Research, pp. 31–40. Springer, Berlin (2002)
22. Durand, C., Fichet, B.: One-t-one correspondences in pyramidal representation: a unified approach. In: Bock, H.H. (ed.) Classification and Related Methods of Data Analysis, pp. 85–90. North-Holland, Amsterdam (1988)
23. Fichet, B.: Data analysis: geometric and algebraic structures. In: Prohorov, Y.A., Sazonov, V.V. (eds.) Proceedings of the First World Congress of the Bernoulli Society (Tachkent, 1986), vol. 2, pp. 123–132. V.N.U. Science Press, Utrecht (1987)
24. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Conceptual Structures: Broadening the Base. Lecture Notes in Computer Science, vol. 2120, pp. 129–142. Springer, Berlin (2001)
25. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**, 241–254 (1967)
26. Mirkin, B., Muchnik, I.: Combinatorial optimization in clustering. In: Du, D.-Z., Pardalos, P. (eds.) Handbook of Combinatorial Optimization, vol. 2, pp. 261–329. Kluwer Academic, Dordrecht (1998)
27. Polaillon, G.: Interpretation and reduction of Galois lattices of complex data. In: Rizzi, A., Vichi, M., Bock, H.-H. (eds.) Advances in Data Science and Classification, pp. 433–440. Springer, Berlin (1998)
28. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht/Boston (1982)

Some Observations on Oligarchies, Internal Direct Sums, and Lattice Congruences

Melvin F. Janowitz

Abstract A set-theoretic abstraction of some deep ideas from lattice theory is presented and discussed. By making use of this abstraction, many results from seemingly disparate disciplines can be examined, proved, and subtle relationships can be discovered among them. Typical applications might involve decision theory when presented with evidence from sources that yield conflicting optimal advice, insights into the internal structure of a finite lattice, and the nature of homomorphic images of a finite lattice. Some needed historical background is provided. (Presented in conjunction with the volume dedicated to the 70th Birthday celebration of Professor Boris Mirkin.) In particular, there is a connection to some early work of Mirkin (On the problem of reconciling partitions. In: Quantitative Sociology, International Perspectives on Mathematical and Statistical Modelling, pp. 441–449. Academic, New York, 1975).

Keywords Oligarchy • Lattice congruence • Simple lattice • Residual mapping

1 Background

A new look at some ideas that are related to a pair of landmark results is presented. First among them is Arrow's Theorem [1]. A connection to simple lattices is motivated and discussed in [20]. Second, there is John von Neumann's famous construction of a continuous generalization of finite dimensional projective geometries, as presented in his 1936–1937 Princeton lectures (see [32]). These are geometries whose subspaces can have any dimension in the real interval [0, 1]. The original definition of a continuous geometry insisted that the underlying lattice

M.F. Janowitz (✉)
DIMACS, Rutgers University, Center/CoRE Building/4th Floor, 96 Frelinghuysen Road, Piscataway, NJ 08854-8018, USA
e-mail: melj@dimacs.rutgers.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_15,

be irreducible in the sense that it has no nontrivial direct product decomposition. There was much interest in developing a version that did not have this restriction. This was especially true in light of Kaplansky's famous result [18] that every complete orthocomplemented modular lattice is a continuous geometry. A subdirect sum representation accomplished this in [21, 22], and at a much later date, a topological representation was produced in [12]. Many other authors pondered this question. F. Maeda's work involved the study of a binary relation which we shall denote as $a \nabla b$. It will turn out that failure of this relation has a connection with congruences of an atomistic lattice, and for that reason it is useful in connection with the study of simple lattices. We shall expand on this connection in the course of our detailed observations. But first some background material is presented in order to provide a framework for the results. We will assume a basic knowledge of lattice theory, but will quickly establish some needed terminology.

We assume the reader is familiar with partial orders. A *lattice* is a partially ordered set L in which every pair a, b of elements has a *least upper bound* $a \vee b$ and a *greatest lower bound* $a \wedge b$. The smallest member of L will be denoted 0 and its largest element 1. A *bounded* lattice has these distinguished members. Thus for any x in such a lattice, it is true that $0 \leq x \leq 1$. A *congruence relation* on L is an equivalence relation Θ such that $a \Theta b$ implies $a \vee c \,\Theta\, b \vee c$ and $a \wedge c \,\Theta\, b \wedge c$ for all $a, b, c \in L$.

Definition 1.1. A *quotient* (denoted s/t) is an ordered pair (s, t) of elements of L with $s \geq t$. Say that $s/t \to u/v$ in one step if for some $w \in L$, $u/v = s \vee w/t \vee w$, or $u/v = s \wedge w/t \wedge w$. Write $s/t \to u/v$ to denote the composition of finitely many relations of the form $x_{i-1}/y_{i-1} \to x_i/y_i$, each in one step, with $x_0/y_0 = s/t$ and the final step ending in $x_n/y_n = u/v$. (Definition from Dilworth [8, p. 349].) To say that $s/t \to u/v$ is to say that the quotient s/t is *weakly projective* onto the quotient u/v. Any congruence Θ is completely determined by the quotients it identifies. The reason for this is that $x \Theta y \iff x \vee y \,\Theta\, x \wedge y$.

For any quotient a/b with $a > b$ here is a formula for the smallest congruence Θ_{ab} that identifies a and b. For $x > y$, $x \Theta_{ab} y$ if and only if there exists a finite chain $x = x_0 > x_1 > \cdots > x_n = y$ such that $a/b \to x_{i-1}/x_i$ for $1 \leq i \leq n$. Though we can keep this in mind, there is a much more concise way of looking at all this when we are dealing with finite lattices. We assume unless otherwise specified that L denotes a finite lattice. A *join-irreducible* member of L is an element $j \in L$ such that $j > 0$ and $j > \bigvee\{x \in L : x < j\}$. Thus j has a unique largest element j_* below it. Every element of L is the join of all join-irreducibles below it, so the structure of L is determined by the set $J(L)$ of all join-irreducibles of L. There is a dual notion $M(L)$ of *meet-irreducibles*. Every $m \in M(L)$ is covered by a unique smallest element m^*, and every element of L is the meet of a family of meet-irreducibles. Note that any congruence Θ of L is completely determined by $\{j \in J(L) : j \Theta j_*\}$, so this gives us another way of thinking about congruences. In particular, we can restrict a congruence to $J(L)$, and just worry about whether quotients of the form j/j_* are collapsed. Of course there are dual notions involving meet-irreducibles. We mention [6, 7, 9, 10] where some of this is discussed and briefly present the items we shall need.

Remark 1.2. The material in this remark is taken from Day [7, pp. 398–399], and [6, p. 72].

- For $p, q \in J(L)$, Day [7] writes qCp to indicate that for some $x \in L$, $q \leq x \vee p$ with $q \not\leq x \vee p_*$, thus forcing $q \not\leq x \vee t$ for any $t < p$. Note that for any congruence Θ, if qCp and $p\Theta p_*$, then $q = q \wedge (p \vee x)\Theta q \wedge (p_* \vee x) < q$ forces $q\Theta q_*$. The idea for the C relation is attributed by Day to material from [28]. *Warning:* Some authors write this relation as pDq or qDp.
- A J-set is a subset $J \subseteq J(L)$ such that $p \in J$ with $qCp \Longrightarrow q \in J$.
- **JSet**(L) is the system of all J-sets of L, ordered by set inclusion.
- There is a natural lattice isomorphism between the congruences on L and $(\mathbf{JSet}(L), \subseteq)$. The association is given by mapping the congruence Θ to $J_\Theta = \{j \in J(L) : j\Theta j_*\}$. Going in the other direction, we can construct the congruence associated with a J-set J by using [9, Lemmas 2.33 and 2.34, p. 40], and defining

$$x\Theta_J y \iff \{a \in J(L) \colon a \leq x,\ a \notin J\} = \{a \in J(L) \colon a \leq y,\ a \notin J\}.$$

  The ordering of the congruences is given by $\Theta_1 \leq \Theta_2 \iff x\Theta_1 y \text{implies } x\Theta_2 y$.
- For each $p \in J(L)$, let Φ_p denote the least congruence that makes p congruent to p_*. Then $J_{\Phi_p} = \{q \in J(L) : q\hat{C}\,p\}$ where $\hat{C}$ is the reflexive transitive closure of C. The reader should observe that J_{Φ_p} is the smallest J-set containing p.
- For $p, q \in J(L)$, it is true that $\Phi_q \leq \Phi_p \iff q \in \Phi_p \iff q\hat{C}\,p$. Thus $\Phi_p = \Phi_q \iff$ both $p\hat{C}\,q$ and $q\hat{C}\,p$.

We mention that Leclerc and Monjardet were independently led to a similar idea in 1990 (see [20, 26] for a discussion of this). For $p, q \in J(L)$, they write $q\delta p$ to indicate that $q \neq p$, and for some $x \in L$, $q \not\leq x$ while $q \leq p \vee x$. They show in [20, Lemma 2], that the relations C and δ coincide if and only if L is atomistic. Here an *atom* of a lattice L with 0 is a minimal element of $L \setminus \{0\}$, and L is *atomistic* if every nonzero element of L is the join of a family of atoms. The dual notions of *dual atoms* (coatoms) and *dual atomistic* (coatomistic) are defined in the expected manner.

2 Results Related to Relations

Think of an underlying finite lattice L, with $J = J(L)$ the set of join-irreducibles of L. Though we are interested in the congruences of L, it turns out to be useful to abstract the situation, see what can be proved, and then later recapture the deep and natural connection with congruences. This idea was already noted by Grätzer and Wehrung in [11]. The situation serves to illustrate one of the most beautiful aspects of mathematics. Looking at an abstraction of a problem can actually simplify proofs and provide more general results. We ask the reader to bear in mind that though we

restrict our attention to finite lattices, we hold open the possibility of establishing a generalization to more general venues.

We begin with some notational conventions. Let J be a finite set, and $R \subseteq J \times J$ a binary relation. For $a \in J$, let $R(a) = \{x \in J : aRx\}$, and for $A \subseteq J$, let $R(A) = \bigcup\{R(a) : a \in A\}$. The relation R^{-1} is defined by $aR^{-1}b \Leftrightarrow bRa$. A subset V of J is called R-closed if $R(V) \subseteq V$, and R^{-1}-closed if $R^{-1}(V) \subseteq V$. It is easily shown that V is R-closed if and only if its complement $J \setminus V$ is R^{-1}-closed. We are interested in the set $\mathcal{V} = \mathcal{V}_R$ of R^{-1}-closed sets, ordered by set inclusion. We chose R^{-1}-closed sets so as to be consistent with the terminology of Remark 1.2. Clearly $(\mathcal{V}, \subseteq)$ is a sublattice of the power set of J, and has the empty set as its smallest member, and J as its largest member. It will be convenient to simply call any $P \in \mathcal{V}$ a J-set to denote the fact that it is R^{-1}-closed. Note that $P \in \mathcal{V}$ has a complement in $\mathcal{V}$ if and only if $J \setminus P \in \mathcal{V}$. Thus P has a complement if and only if it is both R^{-1}-closed and R-closed.

Remark 2.1. The relation R is said to be *reflexive* if jRj for all $j \in J$. It is *transitive* if hRj, jRk together imply that hRk. A relation that is both reflexive and transitive is said to be a *quasiorder*. This is a rather general concept, as every partial order and every equivalence relation is a quasiorder. If the relation R that defines $\mathcal{V}$ is already a quasiorder, then clearly every set of the form $R(a)$ or $R(A)$ is in fact R-closed. Since R^{-1} is also a quasiorder, the same assertion applies to R^{-1}. The relation $R \cap R^{-1}$ is the largest equivalence relation contained in both R and R^{-1}. The least quasiorder containing both R and R^{-1} is denoted $R \vee R^{-1}$, and it is actually also an equivalence relation. The $R \vee R^{-1}$ closed sets are those that are both R and R^{-1} closed.

We could now continue the discussion with a fixed quasiorder R, but we choose instead to have notation that provides an abstract version of Remark 1.2. Accordingly, we take J to be a finite set, but are thinking it as being the join-irreducibles of a finite lattice. A relation R on J is called *irreflexive* if xRx fails for every $x \in J$. We define the relation Δ to be $\{(x, x) : x \in J\}$. We then take R_C to be an irreflexive binary relation on J, and $R_{\hat{C}}$ the reflexive transitive closure of R_C. By this we mean the transitive closure of $\Delta \cup R_C$. Thus $R_{\hat{C}}$ is a *quasiorder* of J. Think of $qR_C\, p$ as the abstraction of qCp, and $qR_{\hat{C}}\, p$ as the abstraction of $q\hat{C}\, p$. We are interested in $\mathcal{V} = \{V \subseteq J : p \in V, qR_C\, p \Longrightarrow q \in V\}$, order it by set inclusion, and call $V \in \mathcal{V}$ a J-set. Note that $\{\emptyset, J\} \subseteq \mathcal{V}$, and that $\mathcal{V}$ is closed under the formation of intersections and unions. Thus $\mathcal{V}$ is a finite distributive lattice. Though R_C is irreflexive, we recall that $R_{\hat{C}}$ is in fact reflexive by its very construction.

Some intuition may be gleaned from a quick look at what happens when $R_{\hat{C}}$ is a partial order. We then write $q \leq p$ to denote the fact that $qR_{\hat{C}}\, p$. We ask what it means for P to be in $\mathcal{V}$. We note that $p \in P, q \leq p$ implies $q \in P$. Thus $\mathcal{V}$ is just the set of order ideals of $(J, \leq)$.

Remark 2.2. Here are some basic facts about $\mathcal{V}$. We remind the reader that each item follows from elementary properties of binary relations; yet, each translates to a known property of congruences on a finite lattice.

1. For each $p \in J$, there is a smallest J-set containing p. We denote this set by V_p, and note that $V_p = \{q \in V : qR_{\hat{C}}\, p\} = R_{\hat{C}}^{-1}(p)$. Thus $V_p \subseteq V_q \iff p \in V_q \iff pR_{\hat{C}}\, q$. The J-sets V_p are clearly the join-irreducibles of $\mathcal{V}$.
2. If $V \in \mathcal{V}$, then $V = \bigcup\{V_p : p \in V\}$.
3. If A is an atom of $\mathcal{V}$, then $p, q \in A \Longrightarrow pR_{\hat{C}}\, q$ and $qR_{\hat{C}}\, p$, so $(p, q) \in R_{\hat{C}} \cap R_{\hat{C}}^{-1}$. Thus A an atom implies $A = V_p$ for any $p \in A$.
4. $R_{\hat{C}}$ is symmetric if and only if $\mathcal{V}$ is a Boolean algebra.

Proof. Suppose first that $R_{\hat{C}}$ is symmetric. We will show that for any $V \in \mathcal{V}$, it is true that $J \setminus V \in \mathcal{V}$. Let $p \in V$ and $q \in J \setminus V$. Suppose $rR_C\, q$. We claim that $r \notin V$. To prove this, we use the symmetry of $R_{\hat{C}}$ to see that $qR_{\hat{C}}\, r$. If $r \in V$, then $qR_{\hat{C}}\, r$ would force $q \in V$, contrary to $q \in J \setminus V$, thus showing that $J \setminus V \in \mathcal{V}$. It follows that $\mathcal{V}$ is complemented, so it is a Boolean algebra.

Suppose conversely that $\mathcal{V}$ is a Boolean algebra. If V_z is an atom of $\mathcal{V}$, then $a \in V_z$ implies $V_a = V_z$, so $a, b \in V_z \Longrightarrow aR_{\hat{C}}\, b$. Thus the restriction of $R_{\hat{C}}$ to V_z is symmetric. What happens if $a \in V_z$ and $b \in J \setminus V_z$? Then both $aR_{\hat{C}}\, b$ and $bR_{\hat{C}}\, a$ must fail. Since J is the union of all atoms of $\mathcal{V}$ it is immediate that $R_{\hat{C}}$ is symmetric. ∎

We note that for congruences on a finite lattice L, this forces the congruence lattice to be a Boolean algebra if and only if the $\hat{C}$ relation on L is symmetric, thus generalizing many known earlier results that have been established for congruences on lattices.

Remark 2.3. It is well known that associated with every quasiordered set there is a homomorphic image that is a partially ordered set. For the quasiorder $R_{\hat{C}}$ that we are considering, here is how the construction goes. We say that $p \sim q$ for $p, q \in V$ if $pR_{\hat{C}}\, q$ and $qR_{\hat{C}}\, p$. Then $\sim$ is an equivalence relation on V, and $\mathcal{V}/\sim$ is a partially ordered set with respect to $\trianglelefteq$ defined by $[p] \trianglelefteq [q]$ if $V_p \subseteq V_q$. One may ultimately show (see Theorem 2.35, p. 41 of [9]) that $(\mathcal{V}, \subseteq)$ is isomorphic to the order ideals of $(\mathcal{V}/\sim, \trianglelefteq)$. If $R_{\hat{C}}$ is symmetric, then it is an equivalence relation. Though one often associates with any equivalence relation its family of partitions, the set $\mathcal{V}$ of J-sets determined by $R_{\hat{C}}$ is most certainly a different object.

If $P \in \mathcal{V}$, we want a formula for the pseudo-complement P^* of P. This is the largest member B of $\mathcal{V}$ such that $P \cap B = \emptyset$. A finite distributive lattice is called a *Stone lattice* if the pseudo-complement of each element has a complement.

Theorem 2.4. *For* $P \in \mathcal{V}$, $P^* = \{q \in J : R_{\hat{C}}^{-1}(q) \cap P = \emptyset\} = J \setminus R_{\hat{C}}(P)$

Proof. We begin by proving the assertion that $\{q \in J : R_{\hat{C}}^{-1}(q) \cap P = \emptyset\} = J \setminus R_{\hat{C}}(P)$. This follows from $\{q \in J : R_{\hat{C}}^{-1}(q) \cap P \neq \emptyset\} = R_{\hat{C}}(P)$. To establish this, note that $q \in R_{\hat{C}}(P) \Leftrightarrow pR_{\hat{C}}\, q$ with $p \in P \Leftrightarrow qR_{\hat{C}}^{-1}\, p$ with $p \in P \Leftrightarrow R_{\hat{C}}^{-1}(q) \cap P \neq \emptyset$. The proof is completed by noting that if $B \in \mathcal{V}$ with $B \cap P = \emptyset$, then $b \in B$, $qR_{\hat{C}}\, b \Longrightarrow q \in B$, so $q \notin P$. This shows that $B \subseteq P^*$. ∎

Lemma 1. *$P \in \mathcal{V}$ has a complement in $\mathcal{V}$ $\iff$ $q \in J \setminus P$, $q_1 R_C q$ implies $q_1 \in J \setminus P$.*

Proof. The condition is just the assertion that $J \setminus P$ is a J-set. ■

Theorem 2.5. *$\mathcal{V}$ is a Stone lattice if and only if $R_{\hat{C}}$ has the property that for each $P \in \mathcal{V}$, $q \notin P^*$ implies that either $q \in P$ or else $q \notin P$ and there exists $q_1 \in P$ such that $q_1 R_{\hat{C}} q$*

Proof. This just applies Lemma 1 to P^*. ■

Here is yet another characterization of when $(\mathcal{V}, \subseteq)$ is a Stone lattice. The result for congruences appears in [13], and the proof we present is just a minor reformulation of the proof that was presented therein. We mention an alternate characterization in the spirit of Dilworth's original approach to congruences that was given in [24]. Note that the arguments in [13] were applied to the set of all prime quotients of a finite lattice, where the argument given here applies to any quasiorder defined on a finite set J. We should also mention earlier and stronger results that appear in [29–31]. So is there anything new in what follows? Only the fact that the proofs can be reformulated for abstract quasiorders.

Theorem 2.6. *$\mathcal{V}$ is a Stone lattice if and only if $R_{\hat{C}}$ has the property that for each $a \in V$ there is one and only one atom V_k of $\mathcal{V}$ such that $V_k \subseteq V_a$.*

Proof. Let $P \in \mathcal{V}$, $a \in J$ with V_k the unique atom of $\mathcal{V}$ that is $\subseteq V_a$. Recall that $V_k \subseteq V_a \iff kR_{\hat{C}}a$.

If $k \notin P$, we let $q \in V$ with $qR_C a$. We will show that $q \notin P$. Let V_j be an atom under V_q. Then $jR_{\hat{C}}q$, $qR_C a$ forces $jR_{\hat{C}}a$. Since there is only one atom under a, we must have $V_j = V_k$, so $kR_{\hat{C}}q$. If $a \in P$, we note that $kR_{\hat{C}}a$ would put $k \in P$, contrary to $k \notin P$. Thus $a \notin P$. Similarly, $q \in P$ produces a contradiction. Thus $q \notin P$ for any $qR_{\hat{C}}a$, and this tells us that $a \in P^*$.

If $k \in P$, then $k \in P^{**}$. Replacing P with P^* in the above argument now shows that $a \in P^{**}$. In any case, $a \in J$ implies $a \in P^* \cup P^{**}$ so P^* and P^{**} are complements.

Now assume that for some $a \in V$ there are two atoms V_j and V_k both contained in V_a. If $a \in V_k^{**}$, then $jR_{\hat{C}}a \implies j \in V_k^{**}$. But $V_j \cap V_k = \emptyset$ implies that $V_j \subseteq V_k^*$, a contradiction. If $a \in V_k^*$, then $kR_{\hat{C}}a$ would put $k \in V_k^*$, contrary to $k \in V_k \subseteq V_k^{**}$. Thus $a \notin V_k^{**} \cup V_k^*$, so V_k^{**} and V_k^* are not complements. ■

Definition 2.7. Let R_C denote an irreflexive binary relation on the finite set J. To say that $\mathcal{V}$ is *subdirectly irreducible* is to say that there is only one atom in $\mathcal{V}$. This is a very old and extremely useful notion in Universal Algebra, and dates back at least to a publication of Birkhoff [2]. It negates the idea of a lattice being subdirectly reducible in the sense that the lattice is a sublattice of a nontrivial direct product of lattices. It just states that there is a nontrivial congruence relation that is contained in any other nontrivial congruence.

The following finite version of a result due to Radeleczki [29–31] now pops out.

Corollary 2.8. *$\mathcal{V}$ is the direct product of subdirectly irreducible factors if and only if for each $a \in J$ there is only one atom $V_k \subseteq V_a$.*

3 The "del" Relation

There is a notion of an internal direct sum of a family of lattices. As a tool toward understanding the internal structure of lattices, there are discussions in [22, pp. 20–25], and [23, pp. 22–24] of what are called internal direct sum decompositions of a lattice with 0. It is shown in both references that the notion of $x \nabla y$ is crucial to this discussion, where $x \nabla y$ indicates that for all $z \in L$, $(x \vee z) \wedge y = z \wedge y$. A more detailed discussion of direct sums occurs in Sect. 4. As we mentioned earlier, this was motivated by investigations into the structure of continuous geometries. Until recently, the author saw no connection between the ∇ relation and congruences on a finite atomistic lattice. But now let's think of what it means for $p \nabla q$ to fail when p, q are distinct atoms. For some $x \in L$, we must have $(p \vee x) \wedge q > x \wedge q$. Then $q \leq p \vee x$, and $q \not\leq 0 \vee x$. Thus qCp. So the fundamental connection for a finite atomistic lattice is given by the fact that for distinct atoms p, q of such a lattice[1],

$$p \nabla q \text{ fails} \iff qCp \tag{1}$$

We mention that this is the reason why $qR_C p$ is taken as the analog of qCp. Having established a connection between the ∇ relation and congruences on a finite atomistic lattice, we look more closely at the del relation on such a lattice. We will restate some pertinent results that were established in [14] back in the 1960s. We mention first that the ∇ relation on arbitrary pairs of elements of a finite atomistic lattice follows quickly from its restriction to pairs of atoms.

Lemma 2. *In a finite atomistic lattice L, $a \nabla b \iff p \nabla q$ for all atoms $p \leq a$ and $q \leq b$.*

Proof. [14, Lemma 6.1, p. 296]. ■

Theorem 3.1. *Let L be a finite atomistic lattice. Every congruence relation Θ of L is the minimal congruence generated by an element s that is* standard *in the sense that $(r \vee s) \wedge t = (r \wedge t) \vee (s \wedge t)$ for all $r, t \in L$. In fact $x \Theta y \iff (x \vee y) = (x \wedge y) \vee s_1$ for some $s_1 \leq s$.*

Proof. Lemma 6.4, p. 297 and Theorem 6.7, p. 298 of [14]. ■

Theorem 3.2. *Let L be a finite dual atomistic lattice. Then $a \nabla b$ in L if and only if $x = (x \vee a) \wedge (x \vee b)$ for all $x \in L$. It follows that $a \nabla b \Longrightarrow b \nabla a$ for all $a, b \in L$.*

[1] Evidently this was known to B. Monjardet and N. Caspard as early as 1995 (Monjardet, private communication).

Proof. Theorem 4.3 of [16]. ■

Definition 3.3. The element z of a bounded lattice L is called *central* if z has a complement z' such that L is isomorphic to $[0, z] \times [0, z']$ via the mapping $x \mapsto (x \wedge z, x \wedge z')$. There is a discussion of this in [22, p. 37].

Theorem 3.4. *Let L be a finite atomistic lattice in which $x \nabla y \implies y \nabla x$ for all $x, y \in L$. Then every congruence on L is the congruence generated by a central element of L. Thus the congruences of L form a finite Boolean algebra.*

Proof. This follows immediately from a stronger result that appeared in Remark 2.2. Nonetheless, we present a direct lattice theoretic proof. By Theorem 3.1, every congruence on L is the minimal one generated by a standard element s. If q is an atom disjoint from s, then $s \nabla q$. By symmetry of ∇, $q \nabla s$. It is immediate that if $t = \bigvee \{ \text{atoms } q \in L : q \not\leq s \}$, then $t \nabla s$. Thus s and t are complements. For any $x \in L$, we note that $x = (x \wedge s) \vee (x \wedge t)$. For if this failed there would be an atom $r \leq x$ such that $r \not\leq (x \wedge s) \vee (x \wedge t)$. But then $r \wedge s = r \wedge t = 0$, a contradiction. Thus s is central [23, Theorem 4.13, p. 18]. ■

Corollary 3.5. *Every finite atomistic lattice in which ∇ is symmetric is a direct product of simple lattices. In particular, this is true for any finite lattice that is both atomistic and dual atomistic.*

Here a lattice is called *simple* if it admits no nontrivial congruence. It follows immediately from Remark 2.2 that finite simple lattices are characterized by the fact that for every pair j, k distinct join-irreducibles, $j \hat{C} k$. A distributive lattice is simple if and only if it has at most two members. One might wonder why Corollary 3.5 leads to a direct product of simple lattices while Proposition 7.2 of [31] leads to a direct product of subdirectly irreducible lattices. The reason is that in the finite case, every congruence relation is the minimal one generated by a central element of the lattice.

It would be interesting to further investigate generalizations of the del relation that are valid for finite lattices that are not atomistic. We outline the start of such a project. For elements a, b of a finite lattice L, we write $a \diamondsuit b$ to denote the fact that they are not comparable (in symbols $a \parallel b$) and for all $x \in L$, $(x \vee a) \wedge b = [x \vee (a \wedge b)] \wedge b$. Note that if $a \wedge b = 0$, this just says that $a \nabla b$. The reason for assuming $a \parallel b$ is that otherwise the assertion that $(x \vee a) \wedge b = [x \vee (a \wedge b)] \wedge b$ is trivially true. In order to obtain a form of separation axiom along the lines of aCb and $a \delta b$, it is convenient to write $a \zeta b$ to indicate that a, b are join-irreducibles with $a \parallel b$ such that $a \leq b \vee x$ and $a \not\leq (a \wedge b) \vee x$ for some $x \in L$. Note that $aCb \implies a \not\leq b$, and $a \zeta b \implies a \parallel b$. For $a \parallel b$, it is evident that $aCb \implies a \zeta b \implies a \delta b$. We might mention that an obvious modification of the proof of [20, Lemma 2] will establish that $\delta = \zeta \iff L$ is atomistic. It is interesting to note that by the same lemma, $\zeta = C$ if and only if L is atomistic. This follows from the fact that if $x < j$ for any join irreducible j, there must then exist a join-irreducible j' with $j' \leq x < j$. Though we have defined ζ and δ to be relations on $J(L)$, it is true that both relations

make sense for any elements of L. We begin our discussion of the diamond relation with a generalization of Theorem 3.2. This result relates equational identities to conditions that involve implications that involve inequalities.

Theorem 3.6. *Let L be a dual atomistic finite lattice. For $a, b \in L$ with $a \parallel b$, the following are equivalent:*

(1) $x \vee (a \wedge b) = (x \vee a) \wedge (x \vee b)$ *for all* $x \in L$.
(2) $a \diamond b$.
(3) $b \leq a \vee x \Longrightarrow b \leq (a \wedge b) \vee x$ *for all* $x \in L$.
(4) $b \leq a \vee d \Longrightarrow b \leq (a \wedge b) \vee d$ *for all dual atoms* d *of* L.

Proof. (1) $\Longrightarrow$ (2) $\Longrightarrow$ (3) $\Longrightarrow$ (4) is obvious, and true for all finite lattices.

(4) $\Longrightarrow$ (1) Suppose (4) holds and $x \vee (a \wedge b) < (x \vee a) \wedge (x \vee b)$. Using the fact that L is dual atomistic, there must exist a dual atom $d \geq x \vee (a \wedge b)$ such that $d \vee [(x \vee a) \wedge (x \vee b] = 1$. Then $d \vee a = d \vee b = 1$. But now $b \leq d \vee a \Longrightarrow b \leq (a \wedge b) \vee d = d$, contrary to $b \vee d = 1$. ■

Corollary 3.7. *Let L be a finite dual atomistic lattice. If $a, b \in L$, then $a \diamond b \iff$ for every dual atom d it is true that $a \wedge b \leq d \Longrightarrow a \leq d$ or $b \leq d$.*

Proof. By applying the Theorem with $x = d$ any dual atom, we see that $a \wedge b \leq d \Longrightarrow a \leq d$ or $b \leq d$. Suppose conversely that the condition holds. For arbitrary $x \in L$, we choose d as in the proof of (4) $\Longrightarrow$ (1) of Theorem 3.6, and apply the condition. ■

Corollary 3.8. *In any dual atomistic finite lattice $a \diamond b$ implies $b \diamond a$.*

Proof. We apply the Theorem to $a \diamond b$, and note that if $x \vee (a \wedge b) = (x \vee a) \wedge (x \vee b)$ for all $x \in L$, then $b \diamond a$. ■

Remark 3.9. Let L be a finite dual atomistic lattice with a, b non-comparable join-irreducibles. Evidently $a \wedge b \leq a_*$ and $a \wedge b \leq b_*$. Suppose $a \diamond b$. Let $x \in L$ be fixed but arbitrary. Using the fact that $x \vee (a \wedge b) = (x \vee a) \wedge (x \vee b)$, we see that

$(x \vee a) \wedge (x \vee b) \leq x \vee a_*$,
$(x \vee a) \wedge (x \vee b) \leq x \vee b_*$, so
$(x \vee a) \wedge (x \vee b) \leq (x \vee a_*) \wedge (x \vee b_*)$.

It is immediate that

$(x \vee a) \wedge (x \vee b) = (x \vee a_*) \wedge (x \vee b_*)$, and so
$(x \vee a) \wedge (x \vee b) = (x \vee a_*) \wedge (x \vee b) = (x \vee a) \wedge (x \vee b_*)$.

Thus $a \wedge (x \vee b) = a \wedge (x \vee b_*)$ and $b \wedge (x \vee a) = b \wedge (x \vee a_*)$.

This shows that $a \leq (x \vee b) \Longrightarrow a \leq (x \vee b_*)$ and $b \leq (x \vee a) \Longrightarrow b \leq (x \vee a_*)$, so both $a C b$ and $b C a$ will fail. Thus for a, b non-comparable join-irreducibles of a finite dual atomistic lattice,

$$a \diamond b \Longrightarrow a C b \text{ and } b C a \text{ must both fail.} \tag{2}$$

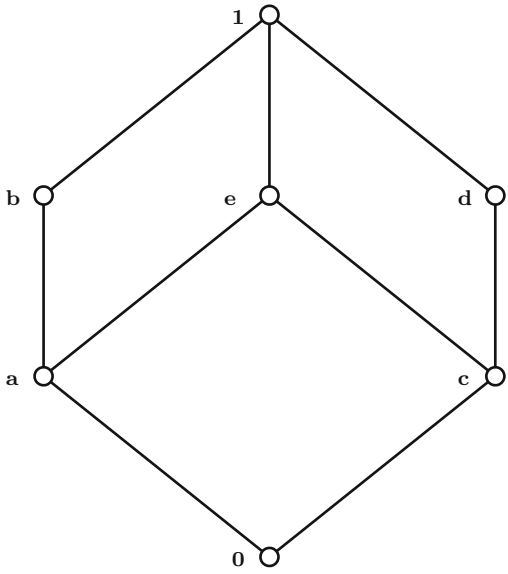


Fig. 1 A dual atomistic lattice

Example 3.10. We present an example to illustrate the approach to congruences on a finite lattice via J-sets. Let L be the five-element non-modular lattice N_5 with coverings $0 < a < b < 1$ and $0 < c < 1$. The join-irreducibles are then a, b, c with $a_* = c_* = 0$ and $b_* = a$. This example is discussed on p. 38 of [10]. There are five J-sets: $\emptyset$, $\{b\}$, $\{a, b\}$, $\{b, c\}$, $\{a, b, c\}$. The J-set $\{b\}$ only produces a single merger of $\{a, b\}$, while the J-set $\{a, b\}$ has two classes $\{0, a, b\}$ and $\{c, 1\}$. Finally, the J-set $\{b, c\}$ has two mergers $\{a, b, 1\}$ and $\{0, c\}$. Note the connection with the fact that L is isomorphic with its dual.

Example 3.11. We next have an example that illustrates what can go wrong for a finite lattice that is not dual atomistic. Let $L = \{0, a, b, c, d, 1\}$ with coverings $0 < a < b < 1$ and $0 < c < d < 1$. The join-irreducibles are $\{a, b, c, d\}$ with $a_* = c_* = 0$, $b_* = a$ and $d_* = c$. Note that $\{b\}$ is a J-set since the merger of b with a is a lattice congruence. Note though that $d \leq b \vee c$, $d \leq b_* \vee c$, and $d \not\leq (d \wedge b) \vee c = c$. Thus $d \zeta b$ does not force d to be a member of the J-set $\{b\}$.

We mention the obvious fact that every result involving finite dual atomistic lattices has a corresponding dual result that is true for finite atomistic lattices.

Example 3.12. In this example, we let L denote the finite lattice depicted in Fig. 1. This lattice was constructed from 2^3 (the Boolean cube) by removal of one atom and all links to that atom. The reader should observe that this lattice is dual atomistic, but not atomistic. The join-irreducibles are a, b, c, d, while the meet-irreducibles are b, d, and e. We leave it to the reader to confirm that the C-relation is given by bCa, bCc, bCd, dCa, dCb, dCc, and that the J-sets are

$$\emptyset, \{b, d\}, \{a, b, d\}, \{c, b, d\}, \{a, b, c, d\}.$$

We now ask what it means for $a \diamond b$ to fail for a, b distinct non-comparable join-irreducibles on a finite dual atomistic lattice L. By Theorem 3.6, this is equivalent to the existence of a dual atom d for which $b \leq a \vee d$ with $b \not\leq (a \wedge b) \vee d$. Thus failure of $a \diamond b$ is equivalent to $b \zeta a$. If follows that the ζ-relation is symmetric. For a finite atomistic lattice, this should be compared to failure of $a \nabla b$ being equivalent to $b C a$. Note the connection with Corollary 15, p. 502 of [26].

4 Internal Direct Sums of a Finite Lattice

Let $S_1, S_2, \ldots, S_n$ be subsets of a lattice L with 0. Following the terminology of Maeda [21], we say that L is the internal direct sum of the S_i if

(1) Each $x \in L$ may be written as $x = \bigvee_{1 \leq i \leq n} x_i$ with $x_i \in S_i$, and
(2) $x \in S_i$, $y \in S_j$ with $i \neq j$ forces $x \nabla y$.

Each S_i is called a *direct summand* of L. There is also a notion of an external direct product of the S_i given by taking the direct product of the family $\{S_i : 1 \leq i \leq n\}$ with the partial order $(a_1, a_2, \ldots, a_n) \leq (b_1, b_2, \ldots, b_n) \iff a_i \leq b_i \ \forall i$. There is then a natural isomorphism between the external direct product of the family S_i and its internal direct sum. It is given by $(a_1, a_2, \ldots, a_n) \longleftrightarrow \bigvee_i a_i$ (see pp. 21–22 of [22]). Having said this, we plan to simplify our notation and identify these two isomorphic entities.

The key item for thinking about all this appears as Theorem 1, p. 1 of [15]. This characterizes direct summands of any lattice L with 0 as central elements of the lattice of ideals of L. For a finite lattice, every ideal is principal, so this tells us that direct summands are generated by the central elements of the lattice. Here is the connection with ∇. By [23, Theorem 4.13, p. 18], in any bounded lattice L, z central in L is equivalent to the existence of an element z' such that $z \nabla z'$, $z' \nabla z$, and $x = (x \wedge z) \vee (x \wedge z')$ for all $x \in L$. The connection with the $\diamond$ relation comes from the fact that in any bounded lattice L,

$$a \diamond b \iff a \nabla b \text{ in } [a \wedge b, 1]. \tag{3}$$

If z is central in L, then clearly $z \vee a$ is central in $[a, 1]$. It would be interesting to investigate the structure of finite lattices where every central member of any filter $[a, 1]$ is of this form. The dual of this condition has been studied for many years, and is called the *relative center property* (RCP). This condition was studied in [17] and examples as well as references were provided therein. The reader might also consult [5] where a connection is given between RCP and congruences in orthomodular lattices. Meaningful examples of what we are discussing may be obtained by just looking at the dual of any lattice that satisfies the RCP. This leads us to investigate the structure of $\{x \in L : x \diamond c\}$ in a finite lattice L. We present a partial result. Further investigation is called for.

Lemma 3. *Let a, b, c be elements of the finite lattice L. Then $a\Diamond c, b\Diamond c \implies (a \vee b)\Diamond c$.*

Proof. Note first that by applying the definition of $\Diamond$ twice, we have

$$(*)\ (a \vee b) \wedge c = [(a \wedge c) \vee b] \wedge c = [(b \wedge c) \vee (a \wedge c)] \wedge c = (a \wedge c) \vee (b \wedge c).$$

Then for any $x \in L$, and again making two uses of the definition of $\Diamond$, followed by an application of $(*)$, we write

$$\begin{aligned}[(a \vee b) \vee x)] \wedge c &= [a \vee (b \vee x)] \wedge c = [(a \wedge c) \vee (b \vee x)] \wedge c \\ &= [b \vee (a \wedge c) \vee x] \wedge c = [(b \wedge c) \vee x \vee (a \wedge c)] \wedge c \\ &= [(a \wedge c) \vee (b \wedge c) \vee x] \wedge c = [[(a \vee b) \wedge c] \vee x] \wedge c. \end{aligned}$$ ∎

Remark 4.1. We mention that any orthomodular as well as any complemented modular lattice that satisfies RCP has the stronger property that the center of any proper interval $[a, b]$ consists of the set of all $(z \vee a) \wedge b$ with z central in L. This is proved using the natural isomorphism of $[a, b]$ with an interval of the form $[0, c]$. It would be interesting if this could be extended to a larger class of relatively complemented lattices. We also mention Theorem 4.4 of [17] where it is shown that for a complete orthomodular lattice RCP is equivalent to e central in $[0, e \vee f]$ with $e \wedge f = 0$ implies $e\nabla f$.

We turn now to a deeper consideration of the structure of a finite atomistic lattice L in which the ∇ relation is symmetric. Recall that for each atom a of L, the smallest J-set containing a is given by $J_a = \{q \in J(L) : q\hat{C}a\}$. We note that J_a generates the smallest congruence relation Θ_a for which a is congruent to 0. By Theorem 3.4, this is the congruence generated by the central element $e(a)$, which is the smallest central element above a. By Theorem 2.5, the pseudocomplement of J_a is given by $J_a^* = J \setminus R_{\hat{C}}(a)$.

In what follows a, b are distinct atoms of L. Since $e(a), e(b)$ are atoms of the center of L, there are only two possibilities: either $e(a) = e(b)$, or $e(a) \wedge e(b) = 0$. For the atoms a, b there are three possibilities: bCa, or bCa fails but $b\hat{C}a$, or $b\hat{C}a$ fails. Recall from Eq. (1) that $a\nabla b$ fails $\iff bCa$.

Lemma 4. $b\hat{C}a \implies e(a) = e(b)$.

Proof. Recall that $e(a), e(b)$ are atoms of the center of L. Suppose bCa and that $e(a) \wedge e(b) = 0$. We know that there is an $x \in L$ such that $b \leq a \vee x$ and $b \not\leq x$. Then

$$b = b \wedge e(b) \leq e(b) \wedge (a \vee x) = (e(b) \wedge a) \vee (e(b) \wedge x) \leq x,$$

a contradiction. Since $\hat{C}$ is the transitive closure of C, it follows that $e(a) = e(b)$, and this completes the proof. ∎

Lemma 5. *Suppose $b\hat{C}a$ fails and $q \in J(L)$ with qCb. Then $q\hat{C}a$ fails. It follows that $b \in J_a^*$, so $e(a) \wedge e(b) = 0$.*

Proof. If $q\hat{C}a$, then by symmetry of ∇, bCq with $q\hat{C}a$ forces $b\hat{C}a$, a contradiction. ■

Theorem 4.2. *Let L be a finite atomistic lattice in which the ∇ relation is symmetric. Then L is either a Boolean lattice, or it is simple with $a\hat{C}b$ for all pairs of atoms a, b or it is a direct sum of such lattices.*

Definition 4.3. In a bounded lattice L, a pair of elements a, b is said to be *perspective* if there is an element x such that $a \vee x = b \vee x$ and $a \wedge x = b \wedge x = 0$. The symbolism for this is $a \sim b$. The transitive closure for a perspective to b is called *a projective* to b.

Lemma 6. *Let L be finite, atomistic, and dual atomistic. If a, b are distinct atoms of L, failure of $a\nabla b$ is equivalent to $a \sim b$. Hence $a \sim b \iff bCa$. This is true also for the dual of L.*

Proof. Suppose $a\nabla b$ fails. There must exist an $x \in L$ such that $x < (x \vee a) \wedge (x \vee b)$. Choose a dual atom $t \geq x$ such that $t \ngeq (x \vee a) \wedge (x \vee b)$. Then $t \ngeq a$ and $t \ngeq b$, so $t \vee a = t \vee b = 1$. Since a, b are atoms, we have $t \wedge a = t \wedge b = 0$. Thus $a \sim b$. The converse is obvious. ■

Theorem 4.4. *Every finite atomistic and dual atomistic lattice is either a Boolean lattice or is a simple lattice in which any pair of atoms is projective and in the relation $\hat{C}$ and dually for dual atoms, or is the internal direct sum of such lattices. In particular this is true for any finite relatively complemented lattice.*

Remark 4.5. We would be remissing if we did not at least mention the connection between a direct summand of a finite lattice and the results from Sect. 2. If we let R_C denote an irreflexive relation on the finite set J, we recall that the J-set P is a direct summand of $\mathcal{V}$ if and only if $J \setminus P$ is an J-set. See Lemma 1.

5 Oligarchies

This entire manuscript has as its original inspiration the appearance of the recent paper [4] by Chambers and Miller. Here is presented a lattice theoretic characterization of when a decision algorithm is an *oligarchy*. An improved result due to Leclerc and Monjardet appears in [20]. The earliest reference the author could find where a lattice theoretic background is provided for a consensus of partitions is the one provided by Mirkin in [25]. This was refined in [19]. See also [27]. We shall be working in a finite lattice L. Intuition may be provided by thinking of L as a model for describing the behavior of a partition of society, or of a partial order or of some concrete decision problem. We shall follow the notation of [20], but will briefly mention here the relevant terminology and notation.

Remark 5.1. A *consensus algorithm* is a mapping $F: L^n \to L$, where L^n is the product of $N = n$ copies of L. We agree to let π denote a typical *profile* $\pi = (x_1, x_2, \ldots, x_n)$ of members of L^n, and $N_x(\pi) = \{i \in N: x \leq x_i\}$. To say that F is *Paretian* is to say that for any atom a, if $N_a(\pi) = N$, then $a \leq F(\pi)$. To say that F is *decisive* is to say that if $N_a(\pi) = N_a(\pi')$ then $a \leq F(\pi) \Leftrightarrow a \leq F(\pi')$. F is *neutral monotone* if for all atoms a, a', and all profiles π, π', $N_a(\pi) \subseteq N_{a'}(\pi')$ implies that if $a \leq F(\pi)$ then $a' \leq F(\pi')$. The constant function that sends every profile π to 0 is denoted F^0.

Finally to say that F is an *oligarchy* is to say that there is a subset M of the indexing set N such that for every profile π, $F(\pi) = \bigwedge\{\pi_i: i \in M\}$. For $x \in L$, we agree to let $\pi_x = (x, x, \ldots, x)$ denote the constant profile having each component x. A mapping $F: L_n \to L$ is called *residual* [3] if it is a meet homomorphism such that $F(\pi_1) = 1$. We mention Theorem 5 of [20] in which the following conditions are shown to be equivalent for any finite simple atomistic lattice L having cardinality greater than 2 and any consensus function $F: L^n \to L$.

Theorem 5.2. *The following conditions are equivalent:*

1. *F is decisive and Paretian.*
2. *F is neutral monotone and is not F^0.*
3. *F is a meet homomorphism and $F(\pi) \geq \bigwedge \pi$ for all profiles π.*
4. *F is a residual map and $F(\pi_a) \geq a$ for every atom a.*
5. *F is an oligarchy.*

We pause to provide a bit of intuitive motivation for the subject at hand. Suppose for the moment that you are in charge of production quotas for a large manufacturing company and that you have an advisory committee consisting of n agents. Each agent i gives you advice in the form of a partition x_i of the space of all possible actions D you might take, and on the basis of these n partitions for π, you must decide on an action $F(\pi)$. The partitions of D may be viewed as a finite simple lattice that is both atomistic and dual atomistic, so we are in a setting where Theorem 5.2 may be applied. Further motivation is provided in [4]. This makes an interesting connection between properties of social choice functions and pure lattice theoretic ideas. It would be interesting to see if this result could be extended to a somewhat broader class of lattice. The key observation is in Corollary 3.5. Making use of this result, we may move from results on a finite simple lattice to results on a direct product of finite simple lattices. Thus we have a characterization of oligarchies on any atomistic finite lattice in which the ∇ relation is symmetric, and in particular for any finite lattice which is both atomistic and dual atomistic. Here specifically is what we have in mind. Let $L_1, L_2, \ldots, L_k$ each denote finite simple lattices having cardinality > 1, and in which the ∇ relation is symmetric. Let F_i be an oligarchy on L_i for each i. Let $L = \Pi_i L_i$ and let F be defined on L by $F(\pi)$ having its ith component the output of F_i applied to the restriction of π to L_i. Then F is a form of generalized oligarchy. It would be of interest to extend Theorem 5.2 to this situation.

6 An Epilogue

We close by reviewing the natural tie between the abstract relation theoretic approach in Sect. 2 and the deep results developed by a number of authors. We especially mention Day [6, 7], and the book by Freese et al. [9].

Remark 6.1. Here then are the main ideas that were covered for the study of congruences on a finite lattice L.

(a) Failure of the ∇ relation on an atomistic lattice and its connection with the C relation.

This is discussed in Sect. 3. See Eq. (1).

(b) The C relation on an arbitrary finite lattice and its abstraction to an irreflexive relation R_C defined on a finite set V.

This is Sect. 2. Remark 2.1 and Theorems 2.4 and 2.6. The abstract formulation can be used to find a generalization of conditions that guarantee that the congruences form a Boolean algebra or a Stone lattice. Noting that $\hat{C}$ is always symmetric for any finite simple lattice, it might be interesting to have an example of a finite simple lattice in which the C relation is not symmetric. It would also be of interest to apply the results more generally to other finite quasiordered sets.

(c) In Sect. 3, a generalization of the ∇-relation was introduced and denoted as $a \diamondsuit b$. There are now three types of separation conditions under consideration. Further work on the connection between these conditions might be appropriate.

Underlying equation $\forall x \in L$	Symbol	Separation condition for some $x \in L$
$(x \vee b) \wedge a = (x \vee b_*) \wedge a$	aCb	$a \leq b \vee x$ and $a \not\leq b_* \vee x$; $a \not\leq b$
$(x \vee b) \wedge a = (x \vee (a \wedge b)) \wedge a$	$a\zeta b$	$a \leq b \vee x$ and $a \leq (a \wedge b) \vee x$; $a \parallel b$
$(x \vee b) \wedge a = x \wedge a$	$a\delta b$	$a \leq b \vee x$ and $a \not\leq x$; $a \neq b$

(d) Section 4 considers internal direct sums of a finite lattice and further explores the connection between the relations ∇ and $\diamondsuit$. As an application, the structure of certain finite atomistic lattices are discussed.

(e) Section 5 gave a quick look at a recent lattice theoretic connection with oligarchies.

Acknowledgments The author wishes to thank Professors Bruno Leclerc, Bernard Monjardet, and Sandor Radeleczki for commenting on earlier versions of the manuscript. Their remarks were a big help. Section 2 especially was revamped because of suggestions from Professor Radeleczki. Thanks are also given to an anonymous referee for many suggestions involving style and clarity of exposition.

References

1. Arrow, K.: Social Choice and Individual Welfare. Wiley, New York (1972)
2. Birkhoff, G.: Subdirect unions in universal algebra. Bull. Am. Math. Soc. **50**, 764–768 (1944)
3. Blyth, T.S., Janowitz, M.F.: Residuation Theory. Pergamon, Oxford (1972)
4. Chambers, C.P., Miller, A.D.: Rules for aggregating information. Soc. Choice Welfare **36**, 75–82 (2011)
5. Chevalier, G.: Around the relative center property in orthomodular lattices. Proc. Am. Math. Soc. **112**, 935–948 (1991)
6. Day, A.: Characterization of finite lattices that are bounded homomorphic images or sublattices of free lattices. Can. J. Math. **31**, 69–78 (1978)
7. Day, A.: Congruence normality: the characterization of the doubling class of convex sets. Algebra Universalsis **31**, 397–406 (1994)
8. Dilworth, R.P.: The structure of relatively complemented lattices. Ann. Math. Second Ser. **51**, 348–359 (1950)
9. Freese, R., Ježek, J., Nation, J.B.: Free Lattices. Mathematical Surveys and Monographs, vol. 42. American Mathematical Society, Providence (1991)
10. Grätzer, G.: The Congruences of a Finite Lattice, A Proof-by-Picture Approach. Birkhäuser, Boston (2006)
11. Grätzer, G., Wehrung, F.: On the number of join-irreducibles in a congruence representation of a finite distributive lattice. Algebra Universalis **49**(2), 165–178 (2003)
12. Harding, J., Janowitz, M.F.: A bundle representation for continuous geometries. Adv. Appl. Math. **19**, 282–293 (1997)
13. Iqbalunisa: On lattices whose lattices of congruence relations are Stone lattices. Fundam. Math. **70**, 315–318 (1971)
14. Janowitz, M.F.: A characterization of standard ideals. Acta Math. Acad. Sci. Hung. **16**, 289–301 (1965)
15. Janowitz, M.F.: A note on normal ideals. J. Sci. Hiroshima Univ. Ser. A-I **30**, 1–8 (1966)
16. Janowitz, M.F.: Section semicomplemented lattices. Math. Z. **108**, 63–76 (1968)
17. Janowitz, M.F.: Separation conditions in relatively complemented lattices. Colloq. Math. **22**, 25–34 (1970)
18. Kaplansky, I.: Any orthocomplemented complete modular lattice is a continuous geometry. Ann. Math. Second Ser. **61**, 524–541 (1955)
19. Leclerc, B.: Efficient and binary consensus function on transitively defined relations. Math. Soc. Sci. **8**, 45–61 (1984)
20. Leclerc, B., Monjardet, B.: Aggregation and Residuation. Order **30**, 261–268 (2013)
21. Maeda, F.: Direct sums and normal ideals of lattices. J. Sci. Hiroshima Univ. Ser. A **14**, 85–92 (1949)
22. Maeda, F.: Kontinuierlichen Geometrien. Springer, Berlin (1958)
23. Maeda, F., Maeda, S.: Theory of Symmetric Lattices. Springer, Berlin (1970)
24. Malliah, C., Bhatta, P.S.: On lattices whose congruences form Stone lattices. Acta Math. Hung. **49**, 385–389 (1987)
25. Mirkin, B.: On the problem of reconciling partitions. In: Quantitative Sociology, International Perspectives on Mathematical and Statistical Modelling, pp. 441–449. Academic, New York (1975)
26. Monjardet, B., Caspard, N.: On a dependence relation in finite lattices. Discrete Math. **165/166**, 497–505 (1997)
27. Nehring, K.: Oligarchies in judgement aggregation. Working paper (2006) http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.389.1952
28. Pudlâk, P., Tuma, J.: Yeast graphs and fermentation of algebraic lattices. In: Colloq. Math, Soc. János Bolyai: Lattice Theory, Szeged, pp. 301–341. North-Holland, Amsterdam (1971)

29. Radeleczki, S.: Some structure theorems for atomistic algebraic lattices. Acta Math. Hung. **86**, 1–15 (2000)
30. Radeleczki, S.: Maeda-type decomposition of CJ-generated algebraic lattices. Southeast Asian Bull. Math. **25**, 503–513 (2001)
31. Radeleczki, S.: The direct decomposition of l-algebras into products of subdirectly irreducible factors. J. Aust. Math. Soc. **75**, 41–56 (2003)
32. von Neumann, J.: Continuous Geometry. Princeton University Press, Princeton (1960/1998)

Thinking Ultrametrically, Thinking p-Adically

Fionn Murtagh

Abstract We describe the use of ultrametric topology and closely associated p-adic number theory in a wide range of fields that all share strong elements of common mathematical and computational underpinnings. These include data analysis, including in the "big data" world of massive and high dimensional data sets; physics at very small scales; search and discovery in general information spaces; and in logic and reasoning.

Keywords Data analytics • Multivariate data analysis • Pattern recognition • Information storage and retrieval • Clustering • Hierarchy • p-Adic • Ultrametric topology • Complexity

1 Introduction

1.1 Hierarchy and Other Symmetries in Data Analysis

On one level this chapter is about symmetries in data, such that the data represent complex phenomena, and the symmetries provide a model for understanding these complex phenomena. Hierarchy gives rise to a rich expanse of symmetries and we will be concerned mostly with hierarchy in this article.

Partitioning a set of observations [47, 70, 71] leads to some very simple symmetries. This is one approach to clustering and data mining. But such approaches, often based on optimization, are not of direct interest to us here. Instead we will pursue the theme pointed to by Simon [69], namely that the notion of hierarchy is fundamental for interpreting data and the complex reality which the data expresses. Our work is

F. Murtagh (✉)
School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
e-mail: fmurtagh@acm.org

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_16,

very different from the marvelous view of the development of mathematical group theory—but viewed in its own right as a complex, evolving system—presented by Foote [24].

Weyl [79] makes the case for the fundamental importance of symmetry in science, engineering, architecture, art and other areas. As a "guiding principle", "Whenever you have to do with a structure-endowed entity ... try to determine its group of automorphisms, the group of those element-wise transformations which leave all structural relations undisturbed. You can expect to gain a deep insight in the constitution of [the structure-endowed entity] in this way. After that you may start to investigate symmetric configurations of elements, i.e. configurations which are invariant under a certain subgroup of the group of all automorphisms; ..." [79, p. 144].

1.2 About This Chapter

Theoretical and applied results that are based on ultrametric topology have been studied in fields such as the following:

- In data analysis, both because of the fitting of tree structures and/or visualizations to data sets, to provide a possible way to present a range of partitions to the user, and also to provide for a genealogical model to be fit to data.
- In physics in order to take account of phenomena at very small spatial and time scales, where it is found that discreteness of structures is represented well by p-adic number systems; and also for any systems that involve movement between discrete states that are characterized by their energy levels. p-Adic number systems can represent ultrametric topology and vice versa.

  It can be added that, as a consequence of applications in physics, the future holds much promise for ultrametric topology-based theory and analysis methods in quantum computing and quantum information theory.
- A further field of use of ultrametric topology arises from being able to show that a considerable number of search and discovery algorithms developed in recent years have an interpretation or vantage point in terms of ultrametric topology.

  Computer programming theory also avails of ultrametrics, for example in order to have a framework for non-monotonic reasoning and for multivalued logic.

This chapter will review the state of the art in these fields and will stress the common aspects of methods and applications.

In Sect. 3, we describe ultrametric topology as an expression of hierarchy.

In Sect. 4, we look at the generalized ultrametric context. This is closely linked with analysis based on lattices.

In Sect. 5, p-adic encoding provides a number theory vantage point on ultrametric topology.

Section 6 deals with application to search and discovery, including work in massive and possibly high dimensional spaces.

2 Backgrounders on Hierarchical Clustering, p-Adic Numbers, Ultrametric Topology

2.1 A Brief Introduction to Hierarchical Clustering

For the reader new to analysis of data a brief introduction is now provided on hierarchical clustering. Along with other families of algorithm, the objective is automatic classification, for the purpose of data mining, or knowledge discovery. Classification, after all, is fundamental in human thinking and machine-based decision making. But we draw attention to the fact that our objective is *unsupervised*, as opposed to *supervised* classification, also known as discriminant analysis or (in a general way) machine learning. So here we are *not* concerned with generalizing the decision-making capability of training data, nor are we concerned with fitting statistical models to data so that these models can play a role in generalizing and predicting. Instead we are concerned with having "data speak for themselves". That this unsupervised objective of classifying data (observations, objects, events, phenomena, etc.) is a huge task in our society is unquestionably true. One may think of situations when precedents are very limited, for instance.

Among families of clustering, or unsupervised classification, algorithms, we can distinguish the following: (a) array permuting and other visualization approaches; (b) partitioning to form (discrete or overlapping) clusters through optimization, including graph-based approaches; and—of interest to us in this article—(c) embedded clusters interrelated in a tree-based way.

For the last-mentioned family of algorithm, agglomerative building of the hierarchy from consideration of object pairwise distances has been the most common approach adopted. For comprehensive background texts, see [33, 34, 45, 80].

2.2 A Brief Introduction to p-Adic Numbers

The real number system and a p-adic number system for given prime, p, are potentially equally useful alternatives. p-Adic numbers were introduced by Kurt Hensel in 1898.

Whether we deal with Euclidean or with non-Euclidean geometry, we are (nearly) always dealing with reals. But the reals start with the natural numbers, and from associating observational facts and details with such numbers we begin the process of measurement. From the natural numbers, we proceed to the rationals, allowing fractions to be taken into consideration.

The following view of how we do science or carry out other quantitative study was proposed by Volovich in 1987 [76, 77]. See also the surveys in [20, 25]. We can always use rationals to make measurements. But they will be approximate, in general. It is better therefore to allow for observables being "continuous, i.e. endow them with a topology". Therefore we need a completion of the field $\mathbb{Q}$ of rationals. To complete the field $\mathbb{Q}$ of rationals, we need Cauchy sequences and this requires a norm on $\mathbb{Q}$ (because the Cauchy sequence must converge, and a norm is the tool used to show this). There is the Archimedean norm such that: for any $x, y \in \mathbb{Q}$, with $|x| < |y|$, then there exists an integer N such that $|Nx| > |y|$. For convenience here, we write: $|x|_\infty$ for this norm. So if this completion is Archimedean, then we have $\mathbb{R} = \mathbb{Q}_\infty$, the reals. That is fine if space is taken as commutative and Euclidean.

What of alternatives? Remarkably all norms are known. Besides the $\mathbb{Q}_\infty$ norm, we have an infinity of norms, $|x|_p$, labelled by primes, p. By Ostrowski's theorem [61] these are all the possible norms on $\mathbb{Q}$. So we have an unambiguous labelling, via p, of the infinite set of non-Archimedean completions of $\mathbb{Q}$ to a field endowed with a topology.

In all cases, we obtain locally compact completions, $\mathbb{Q}_p$, of $\mathbb{Q}$. They are the fields of p-adic numbers. All these $\mathbb{Q}_p$ are continua. Being locally compact, they have additive and multiplicative Haar measures. As such we can integrate over them, such as for the reals.

2.3 Brief Discussion of p-Adic and m-Adic Numbers

We will use p to denote a prime, and m to denote a non-zero positive integer. A p-adic number is such that any set of p integers which are in distinct residue classes modulo p may be used as p-adic digits. (Cf. remark below, at the end of Sect. 5.2, quoting from [29]. It makes the point that this opens up a range of alternative notation options in practice.) Recall that a ring does not allow division, while a field does. m-Adic numbers form a ring; but p-adic numbers form a field. So a priori, 10-adic numbers form a ring. This provides us with a reason for preferring p-adic over m-adic numbers.

We can consider various p-adic expansions:

1. $\sum_{i=0}^{n} a_i p^i$, which defines positive integers. For a p-adic number, we require $a_i \in 0, 1, \ldots p-1$. (In practice: just write the integer in binary form.)
2. $\sum_{i=-\infty}^{n} a_i p^i$ defines rationals.
3. $\sum_{i=k}^{\infty} a_i p^i$ where k is an integer, not necessarily positive, defines the field $\mathbb{Q}_p$ of p-adic numbers.

$\mathbb{Q}_p$, the field of p-adic numbers, is (as seen in these definitions) the field of p-adic expansions.

The choice of p is a practical issue. Indeed, adelic numbers use all possible values of p (see [5] for extensive use and discussion of the adelic number framework). A biotechnology example is considered as follows, by Dragovich and

Dragovich [19] and Khrennikov [38]. Desoxyribonucleic acid (DNA) is encoded using four nucleotides: A, adenine; G, guanine; C, cytosine; and T, thymine. In RNA (ribonucleic acid) T is replaced by U, uracil. In [19] a 5-adic encoding is used, since 5 is a prime and thereby offers uniqueness. In [38] a 4-adic encoding is used, and a 2-adic encoding, with the latter based on 2-digit boolean expressions for the four nucleotides (00, 01, 10, 11). A default norm is used, based on a longest common prefix—with p-adic digits from the start or left of the sequence. (See Sects. 5.3 and 6.3 where a longest common prefix norm or distance is used.)

3 Ultrametric Topology

3.1 Ultrametric Space for Representing Hierarchy

Consider Fig. 1 illustrating the ultrametric distance and its role in defining a hierarchy. An early, influential paper is Johnson [37] and an important survey is that of Rammal et al. [63]. Discussion of how a hierarchy expresses the semantics of change and distinction can be found in [58].

The ultrametric topology was introduced by Krasner [39], the ultrametric inequality having been formulated by Hausdorff in 1934. Essential motivation for the study of this area is provided by Schikhof [66] as follows. Real and complex fields gave rise to the idea of studying any field K with a complete valuation $|.|$ comparable to the absolute value function. Such fields satisfy the "strong triangle inequality" $|x + y| \leq \max(|x|, |y|)$. Given a valued field, defining a totally ordered Abelian (i.e. commutative) group, an ultrametric space is induced through

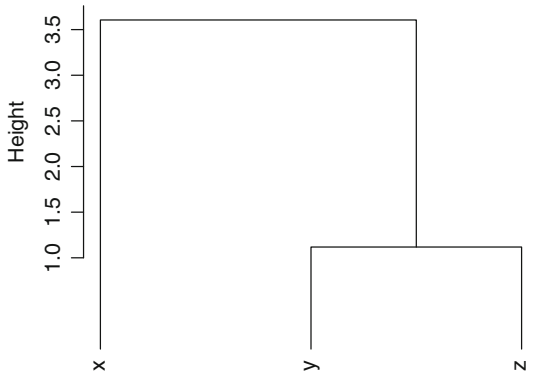


Fig. 1 The strong triangular inequality defines an ultrametric: every triplet of points satisfies the relationship: $d(x, z) \leq \max\{d(x, y), d(y, z)\}$ for distance d. Cf. by reading off the hierarchy, how this is verified for all x, y, z: $d(x, z) = 3.5; d(x, y) = 3.5; d(y, z) = 1.0$. In addition the symmetry and positive definiteness conditions hold for any pair of points

$|x - y| = d(x, y)$. Various terms are used interchangeably for analysis in and over such fields such as p-adic, ultrametric, non-Archimedean, and isosceles. The natural geometric ordering of metric valuations is on the real line, whereas in the ultrametric case the natural ordering is a hierarchical or rooted tree.

3.2 Some Geometrical Properties of Ultrametric Spaces

An ultrametric space is quite different from a metric one. In an ultrametric space everything "lives" on a tree. For various properties that ensue, see [40, Chap. 0, part IV].

In an ultrametric space, all triangles are either isosceles with small base or equilateral. We have here very clear symmetries of shape in an ultrametric topology. These symmetry "patterns" can be used to fingerprint data sets and time series: see [51, 54] for many examples of this.

Some further properties that are studied in [40] are: (a) every point of a circle in an ultrametric space is a centre of the circle. (b) In an ultrametric topology, every ball is both open and closed (termed *clopen*). (c) An ultrametric space is zero-dimensional (see [7, 74]). It is clear that an ultrametric topology is very different from our intuitive, or Euclidean, notions. The most important point to keep in mind is that in an ultrametric space everything "lives" in a hierarchy expressed by a tree.

For an $n \times n$ matrix of positive reals, symmetric with respect to the principal diagonal, to be a matrix of distances associated with an ultrametric distance on X, a sufficient and necessary condition is that a permutation of rows and columns satisfies the following form of the matrix:

1. Above the diagonal term, equal to 0, the elements of the same row are non-decreasing.
2. For every index k, if

$$d(k, k+1) = d(k, k+2) = \cdots = d(k, k+\ell+1)$$

then

$$d(k+1, j) \leq d(k, j) \text{ for } k+1 < j \leq k+\ell+1$$

and

$$d(k+1, j) = d(k, j) \text{ for } j > k+\ell+1$$

Under these circumstances, $\ell \geq 0$ is the length of the section beginning, beyond the principal diagonal, the interval of columns of equal terms in row k.

Table 1 Input data: eight iris flowers characterized by sepal and petal widths and lengths

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
iris1	5.1	3.5	1.4	0.2
iris2	4.9	3.0	1.4	0.2
iris3	4.7	3.2	1.3	0.2
iris4	4.6	3.1	1.5	0.2
iris5	5.0	3.6	1.4	0.2
iris6	5.4	3.9	1.7	0.4
iris7	4.6	3.4	1.4	0.3

From Fisher's iris data [23]

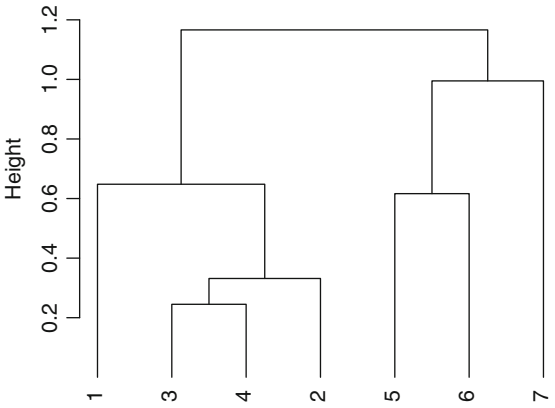


Fig. 2 Hierarchical clustering of seven iris flowers using data from Table 1. No data normalization was used. The agglomerative clustering criterion was the minimum variance or Ward one

Table 2 Ultrametric matrix derived from the dendrogram in Fig. 2

	iris1	iris2	iris3	iris4	iris5	iris6	iris7
iris1	0	0.6480741	0.6480741	0.6480741	1.1661904	1.1661904	1.1661904
iris2	0.6480741	0	0.3316625	0.3316625	1.1661904	1.1661904	1.1661904
iris3	0.6480741	0.3316625	0	0.2449490	1.1661904	1.1661904	1.1661904
iris4	0.6480741	0.3316625	0.2449490	0	1.1661904	1.1661904	1.1661904
iris5	1.1661904	1.1661904	1.1661904	1.1661904	0	0.6164414	0.9949874
iris6	1.1661904	1.1661904	1.1661904	1.1661904	0.6164414	0	0.9949874
iris7	1.1661904	1.1661904	1.1661904	1.1661904	0.9949874	0.9949874	0

To illustrate the ultrametric matrix format, consider the small data set shown in Table 1. A dendrogram produced from this is shown in Fig. 2. From the abscissa height of the lowest node or cluster containing the two terminals, the ultrametric distances, also termed cophenetic distances, matrix can be read off this dendrogram. This is shown in Table 2. Finally a visualization of this matrix, illustrating the ultrametric matrix properties discussed above, is shown in Fig. 3.

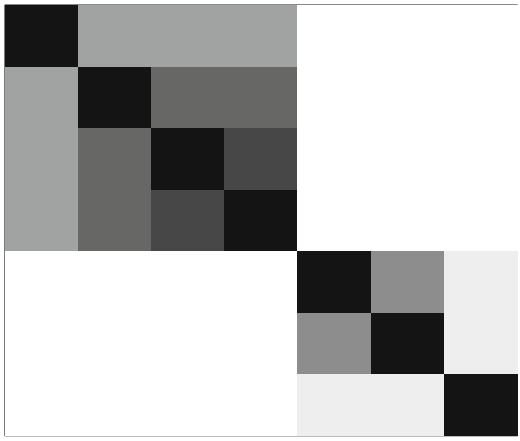


Fig. 3 A visualization of the ultrametric matrix of Table 2, where *bright* or *white* = highest value, and *black* = lowest value

3.3 Clustering Through Matrix Row and Column Permutation

Direct clustering of the data matrix with no changing of the data matrix values—"non-destructive", therefore—also comes under the heading of block model clustering.

Figure 3 shows how an ultrametric distance allows a certain structure to be visible (quite possibly, in practice, subject to an appropriate row and column permuting), in a matrix defined from the set of all distances. A generalization opens up for this sort of clustering-by-visualization scheme. An optimized way to do this was pursued in [43, 44]. Comprehensive surveys of clustering algorithms in this area, including objective functions, visualization schemes, optimization approaches, presence of constraints, and applications, can be found in [42, 72]. See also [17, 50].

For all these approaches, underpinning them are row and column permutations that can be expressed in terms of the permutation group, S_n, on n elements.

4 The Generalized Ultrametric and Formal Concept Analysis

In this section, we consider an ultrametric defined on the power set or join semilattice. Comprehensive background on ordered sets and lattices can be found in [15]. A review of generalized distances and ultrametrics is in [67].

4.1 Link with Formal Concept Analysis

Typically hierarchical clustering is based on a distance (which can be relaxed often to a dissimilarity, not respecting the triangular inequality, and *mutatis mutandis* to a

	v_1	v_2	v_3
a	1	0	1
b	0	1	1
c	1	0	1
e	1	0	0
f	0	0	1

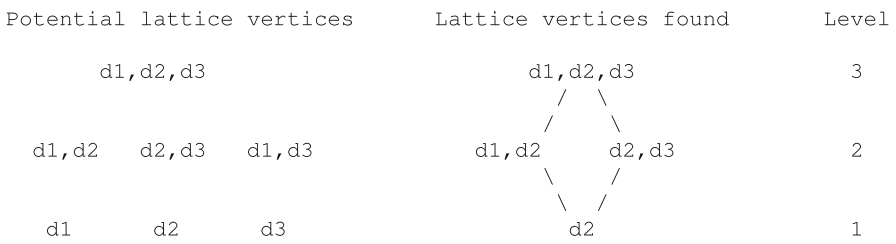


Fig. 4 *Top*: Example data set consisting of five objects, characterized by three boolean attributes. *Bottom*: Lattice corresponding to this data and its interpretation

similarity), defined on all pairs of the object set: $d : X \times X \to \mathbb{R}^+$; i.e., a distance is a positive real value. Usually we require that a distance cannot be 0-valued unless the objects are identical. That is the traditional approach.

A different form of ultrametrization is achieved from a dissimilarity defined on the power set of attributes characterizing the observations (objects, individuals, etc.) X. Here we have: $d : X \times X \longrightarrow 2^J$, where J indexes the attribute (variables, characteristics, properties, etc.) set.

This gives rise to a different notion of distance that maps pairs of objects onto elements of a join semilattice. The latter can represent all subsets of the attribute set, J. That is to say, it can represent the power set, commonly denoted 2^J, of J.

As an example, consider, say, $n = 5$ objects characterized by three boolean (presence/absence) attributes, shown in Fig. 4 (top). Define dissimilarity between a pair of objects in this table as a *set* of 3 components, corresponding to the 3 attributes, such that if both components are 0, we have 1; if either component is 1 and the other 0, we have 1; and if both components are 1, we get 0. This is the simple matching coefficient [36]. We could use, e.g., Euclidean distance for each of the values sought; but we prefer to treat 0 values in both components as signaling a 1 contribution. We get then $d(a, b) = 1, 1, 0$ which we will call `d1,d2`. Then, $d(a, c) = 0, 1, 0$ which we will call `d2`, etc.

We create lattice nodes shown in Fig. 4, as follows.

The set d1,d2,d3 corresponds to: $d(b, e)$ and $d(e, f)$

The subset d1,d2 corresponds to: $d(a, b)$, $d(a, f)$, $d(b, c)$, $d(b, f)$, and $d(c, f)$

The subset d2,d3 corresponds to: $d(a, e)$ and $d(c, e)$

The subset d2 corresponds to: $d(a, c)$

Clusters defined by all pairwise linkage at level ≤ 2:

a, b, c, f

a, c, e

Explanation is as follows: a, b, c, f all share a 1 for attribute v_1 and a 0 for attribute v_2, or vice versa. Then a, c, e share a 1 for attribute v_2 and a 0 for attribute v_3, or vice versa. See this specification of these clusters in the middle part, "Lattice vertices found", of Fig. 4.

Finally:

Clusters defined by all pairwise linkage at level ≤ 3:

a, b, c, e, f

In Formal Concept Analysis [15, 28], it is the lattice itself which is of primary interest. In [36] there is discussion of, and a range of examples on, the close relationship between the traditional hierarchical cluster analysis based on $d : I \times I \rightarrow \mathbb{R}^+$, and hierarchical cluster analysis "based on abstract posets" (a poset is a partially ordered set), based on $d : I \times I \rightarrow 2^J$. The latter, leading to clustering based on dissimilarities, was developed initially in [35].

4.2 Applications of Generalized Ultrametrics

As noted in the previous subsection, the usual ultrametric is an ultrametric distance, i.e. for a set I, $d : I \times I \longrightarrow \mathbb{R}^+$. The generalized ultrametric is also consistent with this definition, where the range is a subset of the power set: $d : I \times I \longrightarrow \Gamma$, where Γ is a partially ordered set. In other words, the *generalized* ultrametric distance is a set. Some areas of application of generalized ultrametrics will now be discussed.

In the theory of reasoning, a monotonic operator is rigorous application of a succession of conditionals (sometimes called consequence relations). However, negation or multiple valued logic (i.e. encompassing intermediate truth and falsehood) requires support for non-monotonic reasoning.

Thus [31]: "Once one introduces negation ... then certain of the important operators are not monotonic (and therefore not continuous), and in consequence the Knaster-Tarski theorem [i.e. for fixed points; see [15]] is no longer applicable to them. Various ways have been proposed to overcome this problem. One such [approach is to use] syntactic conditions on programs ... Another is to consider different operators ... The third main solution is to introduce techniques from topology and analysis to augment arguments based on order ... [the latter include:] methods based on metrics ... on quasi-metrics ... and finally ... on ultrametric spaces".

The convergence to fixed points that are based on a generalized ultrametric system is precisely the study of spherically complete systems and expansive automorphisms discussed in Sect. 5.4 below. As expansive automorphisms we see here again an example of symmetry at work. (Cf. too the quotation from Weyl at the end of Sect. 1.1.)

5 Hierarchy, Ultrametric Topology and the p-Adic Number System

A dendrogram is widely used in hierarchical, agglomerative clustering, and is induced from observed data. By expressing a dendrogram in p-adic terms, we open up a wide range of possibilities for seeing symmetries and attendant invariants.

5.1 p-Adic Numbers and Their Importance

Rizzi [65] considered ultrametrics and ultramines (i.e. an analogous topology for similarities as opposed to dissimilarities, both of which satisfy the strong triangular inequality). He also discussed the representation of ultrametrics and ultramines using p-adic numbers.

The importance of p-adic representation for physics on very small scales has been made by Volovich from the 1980s. See [20, 78]. Such scales are of the order of the Planck length, a fundamental constant (1.6×10^{-35} m). p-Adic description of very large scales has similarly been proposed.

A hierarchy, as a branching process, is a very good means of expressing discrete energy states associated with energy basins requiring at least a requisite quantity of energy to enable a particle to move to another energy basin and possibly, energy level.

Volovich [78] poses the general principle that the fundamental physical laws should be invariant under the change of the number field. Furthermore the p-adic number fields, it is argued, have great benefit at very small scales and at very large scales. This leads to the following ambitious statement: "If these ideas are true then number theory and the corresponding branches of algebraic geometry are ... the ultimate and unified physical theory".

5.2 p-Adic Encoding of a Dendrogram

We will introduce now the one-to-one mapping of clusters (including singletons) in a dendrogram H into a set of p-adically expressed integers (a fortiori, rationals, or $\mathbb{Q}_p$). The field of p-adic numbers is the most important example of ultrametric spaces. Addition and multiplication of p-adic integers, $\mathbb{Z}_p$ (cf. expression in Sect. 2.3), are well defined. Inverses exist and no zero-divisors exist.

A terminal-to-root traversal in a dendrogram or binary rooted tree is defined as follows. We use the path $x \subset q \subset q' \subset q'' \subset \ldots q_{n-1}$, where x is a given object specifying a given terminal, and $q, q', q'', \ldots$ are the embedded classes along this path, specifying nodes in the dendrogram. The root node is specified by the class q_{n-1} comprising all objects.

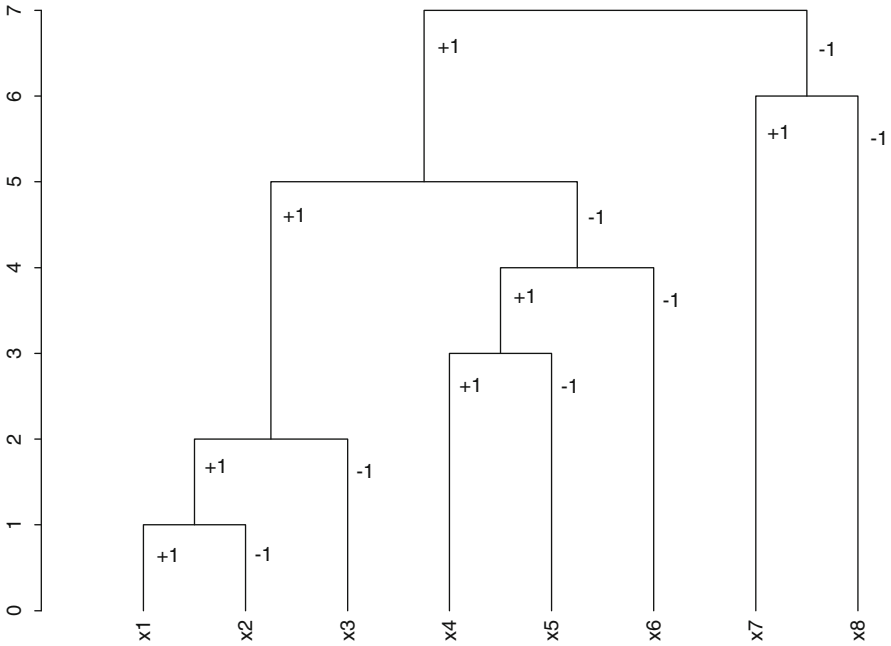


Fig. 5 Labelled, ranked dendrogram on eight terminal nodes, $x_1, x_2, \ldots, x_8$. Branches are labelled $+1$ and -1. Clusters are: $q_1 = (x_1, x_2), q_2 = (x_1, x_2, x_3), q_3 = (x_4, x_5), q_4 = (x_4, x_5, x_6), q_5 = (x_1, x_2, x_3, x_4, x_5, x_6), q_6 = (x_7, x_8), q_7 = (x_1, x_2, \ldots, x_7, x_8)$

A terminal-to-root traversal is the shortest path between the given terminal node and the root node, assuming we preclude repeated traversal (backtrack) of the same path between any two nodes.

By means of terminal-to-root traversals, we define the following p-adic encoding of terminal nodes, and hence objects, in Fig. 5.

$$
\begin{aligned}
x_1 &: +1 \cdot p^1 + 1 \cdot p^2 + 1 \cdot p^5 + 1 \cdot p^7 \\
x_2 &: -1 \cdot p^1 + 1 \cdot p^2 + 1 \cdot p^5 + 1 \cdot p^7 \\
x_3 &: -1 \cdot p^2 + 1 \cdot p^5 + 1 \cdot p^7 \\
x_4 &: +1 \cdot p^3 + 1 \cdot p^4 - 1 \cdot p^5 + 1 \cdot p^7 \\
x_5 &: -1 \cdot p^3 + 1 \cdot p^4 - 1 \cdot p^5 + 1 \cdot p^7 \\
x_6 &: -1 \cdot p^4 - 1 \cdot p^5 + 1 \cdot p^7 \\
x_7 &: +1 \cdot p^6 - 1 \cdot p^7 \\
x_8 &: -1 \cdot p^6 - 1 \cdot p^7
\end{aligned}
\tag{1}
$$

If we choose $p = 2$, the resulting decimal equivalents could be the same: cf. contributions based on $+1 \cdot p^1$ and $-1 \cdot p^1 + 1 \cdot p^2$. Given that the coefficients of

the p^j terms ($1 \leq j \leq 7$) are in the set $\{-1, 0, +1\}$ (implying for x_1 the additional terms: $+0 \cdot p^3 + 0 \cdot p^4 + 0 \cdot p^6$), the coding based on $p = 3$ is required to avoid ambiguity among decimal equivalents.

A few general remarks on this encoding follow. For the labelled ranked binary trees that we are considering (for discussion of combinatorial properties based on labelled, ranked and binary trees, see [49]), we require the labels $+1$ and -1 for the two branches at any node. Of course we could interchange these labels and have these $+1$ and -1 labels reversed at any node. By doing so we will have different p-adic codes for the objects, x_i.

The following properties hold: (a) *Unique encoding:* the decimal codes for each x_i (lexicographically ordered) are unique for $p \geq 3$; and (b) *Reversibility:* the dendrogram can be uniquely reconstructed from any such set of unique codes.

The p-adic encoding defined for any object set can be expressed as follows for any object x associated with a terminal node:

$$x = \sum_{j=1}^{n-1} c_j p^j \text{ where } c_j \in \{-1, 0, +1\} \tag{2}$$

In greater detail we have:

$$x_i = \sum_{j=1}^{n-1} c_{ij} p^j \text{ where } c_{ij} \in \{-1, 0, +1\} \tag{3}$$

Here j is the level or rank (root: $n - 1$; terminal: 1), and i is an object index.

In our example we have used: $c_j = +1$ for a left branch (in the sense of Fig. 5), $c_j = -1$ for a right branch, and $c_j = 0$ when the node is not on the path from that particular terminal to the root.

A matrix form of this encoding is as follows, where $\{\cdot\}^t$ denotes the transpose of the vector.

Let $\mathbf{x}$ be the column vector $(x_1, x_2, \ldots, x_n)^t$.

Let $\mathbf{p}$ be the column vector $(p^1, p^2, \ldots, p^{n-1})^t$.

Define a characteristic matrix C of the branching codes, $+1$ and -1, and an absent or non-existent branching given by 0, as a set of values c_{ij} where $i \in I$, the indices of the object set; and $j \in \{1, 2, \ldots, n-1\}$, the indices of the dendrogram levels or nodes ordered increasingly. For Fig. 5 we therefore have:

$$C = \{c_{ij}\} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{pmatrix} \tag{4}$$

For given level j, $\forall i$, the absolute values $|c_{ij}|$ give the membership function either by node, j, which is therefore read off columnwise or by object index, i, which is therefore read off rowwise.

The matrix form of the p-adic encoding used in Eqs. (2) or (3) is:

$$\mathbf{x} = C\,\mathbf{p} \tag{5}$$

Here, $\mathbf{x}$ is the decimal encoding, C is the matrix with dendrogram branching codes (cf. example shown in expression (4)), and $\mathbf{p}$ is the vector of powers of a fixed prime p.

The tree encoding exemplified in Fig. 5, and defined with coefficients in Eqs. (2) or (3), (4) or (5), with labels $+1$ and -1 was required (as opposed to the choice of 0 and 1, which might have been our first thought) to fully cater for the ranked nodes (i.e. the total order, as opposed to a partial order, on the nodes).

We can consider the objects that we are dealing with to have equivalent integer values. To show that, all we must do is work out decimal equivalents of the p-adic expressions used above for $x_1, x_2, \ldots$. As noted in [29], we have equivalence between: a p-adic number; a p-adic expansion and an element of $\mathbb{Z}_p$ (the p-adic integers). The coefficients used to specify a p-adic number, [29] notes (p. 69), "must be taken in a set of representatives of the class modulo p. The numbers between 0 and $p-1$ are only the most obvious choice for these representatives. There are situations, however, where ther choices are expedient".

We note that the matrix C is used in [14]. A somewhat trivial view of how "hierarchical trees can be perfectly scaled in one dimension" (the title and theme of [14]) is that p-adic numbering is feasible, and hence a one-dimensional representation of terminal nodes is easily arranged through expressing each p-adic number with a real number equivalent.

In [46], what is termed a nest (i.e. cluster nesting) indicator function is defined, based on the set $\{a_w, -b_w, 0\}, a_w, b_w \in \mathbb{R}^+$ in the same way that the set $\{1, -1, 0\}$ is used above for the matrix C. Orthonormality properties of the nest indicator functions are studied.

5.3 *p-Adic Distance on a Dendrogram*

We will now induce a metric topology on the p-adically encoded dendrogram, H. It leads to various symmetries relative to identical norms, for instance, or identical tree distances.

We use the following longest common subsequence, starting at the root: we look for the term p^r in the p-adic codes of the two objects, where r is the lowest level such that both sequences have non-zero, i.e. $+1$ or -1 coefficients, for the p^r term.

Let us look at the set of p-adic codes for $x_1, x_2, \ldots$ above (Fig. 5 and relations (1)), to give some examples of this.

For x_1 and x_2, we find the term we are looking for to be p^1, and so $r = 1$.
For x_1 and x_5, we find the term we are looking for to be p^5, and so $r = 5$.
For x_5 and x_8, we find the term we are looking for to be p^7, and so $r = 7$.

Having found the value r, the distance is defined as p^{-r} [3, 29].

This longest common prefix metric is also known as the Baire distance. In topology the Baire metric is defined on infinite strings [41]. It is more than just a distance: it is an ultrametric bounded from above by 1, and its *infimum* is 0 which is relevant for very long sequences, or in the limit for infinite-length sequences. The use of this Baire metric is pursued in [60] based on random projections [75], and providing computational benefits over the classical $O(n^2)$ hierarchical clustering based on all pairwise distances. This is further discussed in Sect. 6.3.

The longest common prefix metric leads directly to a *p-adic hierarchical classification* (cf. [4]). This is a special case of the "fast" hierarchical clustering to be discussed in Sect. 6.3.

Compared to the longest common prefix metric, there are other related forms of metric, and simultaneously ultrametric. In [27], the metric is defined via the integer part of a real number. In [3], for integers x, y we have: $d(x, y) = 2^{-\text{order}_p(x-y)}$ where p is prime, and $\text{order}_p(i)$ is the exponent (non-negative integer) of p in the prime decomposition of an integer. Furthermore, let $S(x)$ be a series: $S(x) = \sum_{i \in \mathbb{N}} a_i x^i$. ($\mathbb{N}$ is the set of natural numbers.) The order of $S(x)$ is the rank of its first non-zero term: $\text{order}(S) = \inf\{i : i \in \mathbb{N}; a_i \neq 0\}$. (The series that is all zero is of order infinity.) Then the ultrametric similarity between series is: $d(S, S') = 2^{-\text{order}(S-S')}$.

5.4 Scale-Related Symmetry

Scale-related symmetry is very important in practice. In this subsection we introduce an operator that provides this symmetry. We also term it a dilation operator, because of its role in the wavelet transform on trees (see [55] for discussion and examples). This operator is p-adic multiplication by $1/p$.

Consider the set of objects $\{x_i | i \in I\}$ with its p-adic coding considered above. Take $p = 2$. (Non-uniqueness of corresponding decimal codes is not of concern to us now, and taking this value for p is without any loss of generality.) Multiplication of $x_1 = +1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^5 + 1 \cdot 2^7$ by $1/p = 1/2$ gives: $+1 \cdot 2^1 + 1 \cdot 2^4 + 1 \cdot 2^6$. Each level has decreased by one, and the lowest level has been lost. Subject to the lowest level of the tree being lost, the form of the tree remains the same. By carrying out the multiplication-by-$1/p$ operation on all objects, it is seen that the effect is to rise in the hierarchy by one level.

Let us call product with $1/p$ the operator A. The effect of losing the bottom level of the dendrogram means that either (1) each cluster (possibly singleton) remains the same; or (2) two clusters are merged. Therefore the application of A to all q implies a subset relationship between the set of clusters $\{q\}$ and the result of applying A, $\{Aq\}$.

Repeated application of the operator A gives Aq, A^2q, A^3q, ... Starting with any singleton, $i \in I$, this gives a path from the terminal to the root node in the tree. Each such path ends with the null element, which we define to be the p-adic encoding corresponding to the root node of the tree. Therefore the intersection of the paths equals the null element.

Benedetto and Benedetto [1, 2] discuss A as an expansive automorphism of I, i.e. form-preserving, and locally expansive. Some implications [2] of the expansive automorphism follow. For any q, let us take $q, Aq, A^2q, \ldots$ as a sequence of open subgroups of I, with $q \subset Aq \subset A^2q \subset \ldots$, and $I = \bigcup\{q, Aq, A^2q, \ldots\}$. This is termed an inductive sequence of I, and I itself is the inductive limit [64, p. 131].

Each path defined by application of the expansive automorphism defines a spherically complete system [27, 66, 74], which is a formalization of well-defined subset embeddedness. Such a methodological framework finds application in multi-valued and non-monotonic reasoning, as noted in Sect. 4.2.

6 Exploiting Ultrametric Embeddings for Search and Discovery

6.1 Remarkable Symmetries in Very High Dimensional Spaces

In the work of [62, 63] it was shown how as ambient dimensionality increased distances became more and more ultrametric. That is to say, a hierarchical embedding becomes more and more immediate and direct as dimensionality increases. A better way of quantifying this phenomenon was developed in [51]. What this means is that there is inherent hierarchical structure in high dimensional data spaces.

It was shown experimentally in [51, 62, 63] how points in high dimensional spaces become increasingly equidistant with increase in dimensionality. Both [30] and [18] study Gaussian clouds in very high dimensions. The latter finds that "not only are the points [of a Gaussian cloud in very high dimensional space] on the convex hull, but all reasonable-sized subsets span faces of the convex hull. This is wildly different than the behavior that would be expected by traditional low-dimensional thinking".

That very simple structures come about in very high dimensions is not as trivial as it might appear at first sight. Firstly, even very simple structures (hence with many symmetries) can be used to support fast and perhaps even constant time worst case proximity search [51]. Secondly, as shown in the machine learning framework by Hall et al. [30], there are important implications ensuing from the simple high dimensional structures. Thirdly, [56] shows that very high dimensional clustered data contain symmetries that in fact can be exploited to "read off" the clusters in a computationally efficient way. Fourthly, following [16], what we might want to look for in contexts of considerable symmetry are the "impurities" or small irregularities that detract from the overall dominant picture.

Table 3 Typical results, based on 300 sampled triangles from triplets of points

No. points	Dimen.	Isosc.	Equil.	UM
Uniform				
100	20	0.10	0.03	0.13
100	200	0.16	0.20	0.36
100	2000	0.01	0.83	0.84
100	20000	0	0.94	0.94
Hypercube				
100	20	0.14	0.02	0.16
100	200	0.16	0.21	0.36
100	2000	0.01	0.86	0.87
100	20000	0	0.96	0.96
Gaussian				
100	20	0.12	0.01	0.13
100	200	0.23	0.14	0.36
100	2000	0.04	0.77	0.80
100	20000	0	0.98	0.98

For uniform, the data are generated on $[0, 1]^m$; hypercube vertices are in $\{0, 1\}^m$, and for Gaussian on each dimension, the data are of mean 0, and variance 1. Dimen. is the ambient dimensionality. Isosc. is the number of isosceles triangles with small base, as a proportion of all triangles sampled. Equil. is the number of equilateral triangles as a proportion of triangles sampled. UM is the proportion of ultrametricity-respecting triangles ($=1$ for all ultrametric)

See Table 3 exemplifying the change of topological properties as ambient dimensionality increases. It behoves us to exploit the symmetries that arise when we have to process very high dimensional data.

6.2 Partial Ultrametric Embedding

In [57], we discuss permutation representations of a data stream. Since hierarchies can also be represented as permutations, there is a ready way to associate data streams with hierarchies. In fact, early computational work on hierarchical clustering used permutation representation to great effect (cf. [68]).

To analyse data streams in this way, in [54] we develop an approach to ultrametric embedding of time-varying signals, including biomedical, meteorological, financial and other. As opposed to the classical way of inducing a hierarchy, through use of an agglomerative hierarchical clustering algorithm, we look for the ultrametric relationship—the strong triangular inequality—and, when found, count such particular cases of adherence to inherent hierarchical properties in the data. The most non-ultrametric time series are found to be chaotic ones. Eyegaze trace data was found to be remarkably high in ultrametricity, which are likely to be due to extreme saccade movements. Some initial questions were raised in that work [54] in regard to the EEG data used, for sleeping, petit mal and irregular epilepsy cases.

This work has been pursued by Khrennikov and his colleagues in modelling multi-agent systems. See [22]. Furthermore this work uses Bose–Einstein and Fermi–Dirac statistical distributions (derived from quantum statistics of energy states of bosons and fermions, i.e. elementary particles with integer, and half odd integer, spin). In [21, 22] multi-agent behaviours are modelled using such energy distributions. The framework is an urn model, where balls can move, with loss of energy over time, and with possibilities to receive input energy, but potentially shared with other balls. See the cited works for a full description of the Monte Carlo system set up. Sequences of actions (and moves), viz. their histories, are coded such that triangle properties can be investigated (cf. also [54]). That leads to a characterization of how ultrametrically embeddable the data is, ab initio (and not through imposing any hierarchical or other structure on the data with retrospective goodness of fit assessment). In [22], the case is presented for such analysis of behavioural histories being important for study of social and economic complexity.

Quantum statistical distributions have been noted in the foregoing work [21, 22]. van Rijsbergen [73] has set out various ways in which a quantum physics formalism makes clearer what is being done in information retrieval and in data analysis generally.

The quantifying of the inherent ultrametric content of text, and finding that some are much more inherently hierarchical than others, was pursued in [59]. As data, the following were used: tales from the Brothers Grimm, Jane Austen novels, dream reports, air accident reports, and James Joyce's Ulysses.

6.3 Ultrametric Baire Space and Distance

A Baire space consists of countably infinite sequences with a metric defined in terms of the longest common prefix: the longer the common prefix, the closer a pair of sequences. This longest common prefix metric allows us to define the Baire distance [11, 48, 60]. In this description of the Baire distance, we consider to begin with scalar or univariate values. Below we will generalize to multivariate data such as in the case, for example, of documents with presence or absence, or frequencies of occurrence, on a term set or some other features.

Take the longest common prefixes at issue here as coming from precision of any value. Without loss of generality, take these values as decimal, i.e. base 10, or m-adic with $m = 10$. We take x and y to be bounded by 0 and 1. Each of them is of some precision, and we take the integer $|K|$ to be the maximum precision.

Thus we consider ordered sets x_k and y_k for $k \in K$, or, we will write, for $k = 1, 2, \ldots, |K|$. The cardinality of the set K is the precision with which a number, x, is measured. So, x_k with $k = 1$ is the first decimal place of precision; with $k = 2$, we have the second decimal place; . . . ; and with $k = |K|$ we have the $|K|$th decimal place.

Consider as examples $x = 0.478$; and $y = 0.472$. Start from the first decimal position. For $k = 1$, we find $x_1 = y_1 = 4$. For $k = 2$, $x_2 = y_2 = 7$. But for $k = 3$, $x_3 \neq y_3$.

We now introduce the following distance (case of x and y, with 1 attribute, hence unidimensional, and where subscript k is the digit of precision):

$$d_{\mathrm{B}}(x, y) \equiv d_{\mathrm{B}}(x_K, y_K) = \begin{cases} 1 & \text{if } x_1 \neq y_1 \\ \inf\ 10^{-k} & x_k = y_k, \quad 1 \leq k \leq |K| \end{cases} \tag{6}$$

We call this d_{B} value Baire distance, which can be shown to be an ultrametric [51–54, 60] distance. In the properties of a metric we generally have $d(x, y) = 0$ iff$x = y$ whereas for the Baire distance this reflexivity property is relaxed by having the 0 value replaced by the definably minimal value.

When dealing with binary data, 2 is a convenient base. In definition (6) we used a base of 10 for ease of coding when working with real numbers.

It is seen that this distance splits a unidimensional string of decimal values into a 10-way hierarchy, in which each leaf is associated with a grid cell. From Eq. (6) we can read off the distance between points assigned to the same grid cell. All pairwise distances of points assigned to the same cell are the same.

Relative to agglomerative hierarchical clustering, the Baire-based hierarchy is such that each node of this tree is associated with a grid (more strictly, in what we have described, interval) cell. Cell assignments at a particular level can be used to count the number of values x, y that are associated with that cell, and these counts define local density. As we have described the inducing of a hierarchy, this has been in a top-down manner (cf. how agglomerative hierarchical clustering, in that it is agglomerative, is consequently bottom-up). It follows from this algorithm that we can read the hierarchy off the data in a single scan, by having the target data structure—here, a regular 10-way tree—and assigning each value to all its appropriate nodes in the tree. For ease of characterizing this tree, or hierarchical clustering, we refer to it as a *Baire tree* or *Baire hierarchy*. The minimum Baire distance corresponds to a partial match of the values at each level [13].

For data with higher dimensionality, random projections can be used. Random projection is simple. Forming the random matrix R and projecting the $d \times N$ data matrix X into the k dimensions is of order $O(dkN)$. If X is sparse with c non-zero entries per column, the complexity is of order $O(ckN)$.

In fact random projection can be seen as a class of hashing function. Hashing is much faster than alternative methods because it avoids the pairwise comparisons required for partitioning and classification. If two points (p, q) are close, they will have a very small $\|p - q\|$ (Euclidean metric) value; and they will hash to the same value with high probability; if they are distant, they should collide with small probability.

Clustering using the Baire distance has been successfully applied to areas such as chemoinformatics [60], astronomy [12] and text retrieval [10].

In [60], this principle of binning data is used on a large, high dimensional chemoinformatics data set. The application of merging databases of chemical

compounds is important. In [60] 1.2 million compounds were used, characterized using a particular coding scheme by 1052-valued presence/absence vector. Use of the Baire distance and the associated hierarchical clustering were compared in detail with k-means partitioning (through partitions derived from the hierarchical clustering).

We studied stability of results, and effectiveness relative to other clustering methods, in particular k-means partitioning, in [12]. The main domain of application in that work was astronomy, and in particular clustering of redshifts in order to facilitate regression of (more expensively observed but better quality) spectroscopic redshifts on (more easily observed but with less signal resolution) photometric redshifts.

6.4 Approximating an Ultrametric for Similarity Metric Space Searching

In [51] we show that, in much work over the years, nearest neighbour searching has been made more efficient through the use of more easily determined feasibility bounds. An early example is Fukunaga and Narendra [26], a chapter review is in [50], and a survey can be found in [9]. Rendering given distances as ultrametric is a powerful way to facilitate nearest neighbour searching. Furthermore "stretching the triangular inequality" (a phrase used by [8]) so that it becomes the strong triangular inequality, or ultrametric inequality, gives a unifying view of some algorithms of this type.

Fast nearest neighbour finding often makes use of pivots to establish bounds on points to be searched, and points to be bypassed as infeasible [6, 9].

A full discussion can be found in [51]. Fast nearest neighbour searching in metric spaces often appeals to heuristics. The link with ultrametric spaces gives rise instead to a unifying view.

Hjaltason and Samet [32] discuss heuristic nearest neighbour searching in terms of embedding the given metric space points in lower dimensional spaces. From our discussion in this section, we see that there is evidently another alternative direction for facilitating fast nearest neighbour searching: viz., taking the metric space as an ultrametric one, and if it does not quite fit this perspective, then "stretch" it (transform it locally) so that it does so.

7 Conclusions

There are many exciting perspectives opened up by our work on the theme of symmetry-finding through hierarchy in very large data collections, with insights and perspectives from many application domains that are data-based and motivated, and indeed driven, by complex problem-solving.

"My thesis has been that one path to the construction of a nontrivial theory of complex systems is by way of a theory of hierarchy". Thus Simon [69, p. 216]. We have noted symmetry in many guises in the representations used, in the transformations applied, and in the transformed outputs. These symmetries are non-trivial too, in a way that would not be the case were we simply to look at classes of a partition and claim that cluster members were mutually similar in some way. We have seen how the p-adic or ultrametric framework provides significant focus and commonality of viewpoint.

Furthermore we have highlighted the computational scaling properties of our algorithms. They are fully capable of addressing the data and information deluge that we face and providing us with the best interpretative and decision-making tools. The full elaboration of this last point is to be sought in each and every application domain, and face to face with old and new problems.

References

1. Benedetto, R.L.: Examples of wavelets for local fields. In: Larson, D., Heil, C., Jorgensen, P. (eds.) Wavelets, Frames, and Operator Theory, Contemporary Mathematics, American Mathematical Society (Providence, RI) vol. 345, pp. 27–47 (2004)
2. Benedetto, J.J., Benedetto, R.L.: A wavelet theory for local fields and related groups. J. Geom. Anal. **14**, 423–456 (2004)
3. Benzécri, J.-P.: L'Analyse des Données. Tome I. Taxinomie, 2nd edn. Dunod, Paris (1979)
4. Bradley, P.E.: Mumford dendrograms. Comput. J. **53**, 393–404 (2010)
5. Brekke, L., Freund, P.G.O.: p-Adic numbers in physics. Phys. Rep. **233**, 1–66 (1993)
6. Bustos, D., Navarro, G., Chávez, E.: Pivot selection techniques for proximity searching in metric spaces. Pattern Recognit. Lett. **24**, 2357–2366 (2003)
7. Chakraborty, P.: Looking through newly to the amazing irrationals. Technical report (2005). arXiv: math.HO/0502049v1
8. Chávez, E., Navarro, G.: Probabilistic proximity search: fighting the curse of dimensionality in metric spaces. Inf. Process. Lett. **85**, 39–46 (2003)
9. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L.: Searching in metric spaces. ACM Comput. Surv. **33**(3), 273–321 (2001)
10. Contreras, P.: Search and Retrieval in Massive Data Collections. Ph.D. thesis, Royal Holloway, University of London (2011)
11. Contreras, P., Murtagh, F.: Evaluation of hierarchies based on the longest common prefix, or Baire, metric, 2007. In: Classification Society of North America (CSNA) Meeting, University of Illinois, Urbana-Champaign, IL (2007)
12. Contreras, P., Murtagh, F.: Fast, linear time hierarchical clustering using the Baire metric. J. Classif. **29**, 118–143 (2012)
13. Contreras, P., Murtagh, F.: Linear time Baire hierarchical clustering for enterprise information retrieval. Int. J. Softw. Inform. **6**(3), 363–380 (2012)
14. Critchley, F., Heiser, W.: Hierarchical trees can be perfectly scaled in one dimension. J. Classif. **5**, 5–20 (1988)
15. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order, 2nd edn. Cambridge University Press, Cambridge (2002)
16. Delon, F.: Espaces ultramétriques. J. Symb. Logic **49**, 405–502 (1984)
17. Deutsch, S.B., Martin, J.J.: An ordering algorithm for analysis of data arrays. Oper. Res. **19**, 1350–1362 (1971)

18. Donoho, D.L., Tanner, J.: Neighborliness of randomly-projected simplices in high dimensions. Proc. Natl. Acad. Sci. **102**, 9452–9457 (2005)
19. Dragovich, B., Dragovich, A.: p-Adic modelling of the genome and the genetic code. Comput. J. **53**, 432–442 (2010)
20. Dragovich, B., Khrennikov, A.Yu., Kozyrev, S.V., Volovich, I.V.: On p-adic mathematical physics. p-Adic Numbers Ultrametric Anal. Appl. **1**, 1–27 (2009)
21. Ezhov, A.A., Khrennikov, A.Yu.: On ultrametricity and a symmetry between Bose-Einstein and Fermi-Dirac systems. In: AIP Conferences Proceedings 826, p-Adic Mathematical Physics, 2nd International Conference, pp. 55–64. American Institute of Physics, Melville (2006)
22. Ezhov, A.A., Khrennikov, A.Yu., Terentyeva, S.S.: Indications of a possible symmetry and its breaking in a many-agent model obeying quantum statistics. Phys. Rev. E, **77** (2008). Article number 031126
23. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**, 179–188 (1936)
24. Foote, R.: Mathematics and complex systems. Science **318**, 410–412 (2007)
25. Freund, P.G.O.: p-Adic strings and their applications. In: Rakic, Z., Dragovich, B., Khrennikov, A., Volovich, I. (eds.) Proceedings of 2nd International Conference on p-Adic Mathematical Physics, pp. 65–73. American Institute of Physics, Melville (2006)
26. Fukunaga, K., Narendra, P.M.: A branch and bound algorithm for computing k-nearest neighbors. IEEE Trans. Comput. **C-24**, 750–753 (1975)
27. Gajić, L.: On ultrametric space. Novi Sad J. Math. **31**, 69–71 (2001)
28. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999) [Formale Begriffsanalyse. Mathematische Grundlagen. Springer, Berlin (1996)]
29. Gouvêa, F.Q.: p-Adic Numbers: An Introduction. Springer, Berlin (2003)
30. Hall, P., Marron, J.S., Neeman, A.: Geometric representation of high dimensional, low sample size data. J. R. Stat. Soc. B **67**, 427–444 (2005)
31. Hitzler, P., Seda, A.K.: The fixed-point theorems of Priess-Crampe and Ribenboim in logic programming. Fields Inst. Commun. **32**, 219–235 (2002)
32. Hjaltason, G.R., Samet, H.: Properties of embedding methods for similarity searching in metric spaces. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 530–549 (2003)
33. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Upper Saddle River (1988)
34. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**, 264–323 (1999)
35. Janowitz, M.F.: An order theoretic model for cluster analysis. SIAM J. Appl. Math. **34**, 55–72 (1978)
36. Janowitz, M.F.: Cluster analysis based on abstract posets. Technical report (2005–2006)
37. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**, 241–254 (1967)
38. Khrennikov, A.Yu.: Gene expression from polynomial dynamics in the 2-adic information space. Technical report (2009)
39. Krasner, M.: Nombres semi-réels et espaces ultramétriques. C. R. Acad. Sci., Tome II **219**, 433 (1944)
40. Lerman, I.C.: Classification et Analyse Ordinale des Données. Dunod, Paris (1981)
41. Levy, A.: Basic Set Theory. Dover, Mineola (2002) [Springer, 1979]
42. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. Comput. Biol. Bioinform. **1**, 24–45 (2004)
43. March, S.T.: Techniques for structuring database records. Comput. Surv. **15**, 45–79 (1983)
44. McCormick, W.T., Schweitzer, P.J., White, T.J.: Problem decomposition and data reorganization by a clustering technique. Oper. Res. **20**, 993–1009 (1982)
45. Mirkin, B.: Mathematical Classification and Clustering. Kluwer, Dordrecht (1996)
46. Mirkin, B.: Linear embedding of binary hierarchies and its applications. In: Mirkin, B., McMorris, F., Roberts, F., Rzhetsky, A. (eds.) Mathematical Hierarchies and Biology. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 37, pp. 331–356. American Mathematical Society, Providence (1997)

47. Mirkin, B.: Clustering for Data Mining. Chapman and Hall/CRC, Boca Raton (2005)
48. Mirkin, B., Fishburn, P.: Group Choice. V.H. Winston, Washington (1979)
49. Murtagh, F.: Counting dendrograms: a survey. Discrete Appl. Math. **7**, 191–199 (1984)
50. Murtagh, F.: Multidimensional Clustering Algorithms. Physica-Verlag, Heidelberg/Vienna (1985)
51. Murtagh, F.: On ultrametricity, data coding, and computation. J. Classif. **21**, 167–184 (2004)
52. Murtagh, F.: Quantifying ultrametricity. In: Antoch, J. (ed.) COMPSTAT 2004 – Proceedings in Computational Statistics, pp. 1561–1568. Springer, Berlin (2004)
53. Murtagh, F.: Thinking ultrametrically. In: Banks, D., House, L., McMorris, F.R., Arabie, P., Gaul, W. (eds.) Classification, Clustering, and Data Mining Applications. Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), pp. 3–14. Springer, Berlin (2004)
54. Murtagh, F.: Identifying the ultrametricity of time series. Eur. Phys. J. B **43**, 573–579 (2005)
55. Murtagh, F.: The Haar wavelet transform of a dendrogram. J. Classif. **24**, 3–32 (2007)
56. Murtagh, F.: The remarkable simplicity of very high dimensional data: application to model-based clustering. J. Classif. **26**, 249–277 (2009)
57. Murtagh, F.: Symmetry in data mining and analysis: a unifying view based on hierarchy. Proc. Steklov Inst. Math. **265**, 177–198 (2009)
58. Murtagh, F.; The correspondence analysis platform for uncovering deep structure in data and information (6th Annual Boole Lecture). Comput. J. **53**, 304–315 (2010)
59. Murtagh, F.: Ultrametric model of mind, II: application to text content analysis. p-Adic Numbers Ultrametric Anal. Appl. **4**(3), 207–221 (2012)
60. Murtagh, F., Downs, G., Contreras, P.: Hierarchical clustering of massive, high dimensional data sets by exploiting ultrametric embedding. SIAM J. Sci. Comput. **30**, 707–730 (2008)
61. Ostrowski, A.: Über einige Lösungen der Funktionalgleichung $\phi(x)\cdot\phi(y)-\phi(xy)$. Acta Math. **41**, 271–284 (1918)
62. Rammal, R., Angles d'Auriac, J.C., Doucot, B.: On the degree of ultrametricity. J. Phys. Lett. **46**, L-945–L-952 (1985)
63. Rammal, R., Toulouse, G., Virasoro, M.A.: Ultrametricity for physicists. Rev. Mod. Phys. **58**(3), 765–788 (1986)
64. Reiter, H., Stegeman, J.D.: Classical Harmonic Analysis and Locally Compact Groups, 2nd edn. Oxford University Press, Oxford (2000)
65. Rizzi, A.: Ultrametrics and p-adic numbers. In: Gaul, W., Opitz, O., Schader, M. (eds.) Data Analysis: Scientific Modeling and Practical Application, pp. 325–324. Springer, Berlin (2000)
66. Schikhof, W.H.: Ultrametric Calculus. Cambridge University Press, Cambridge (1984) [Chaps. 18–21]
67. Seda, A.K., Hitzler, P.: Generalized distance functions in the theory of computation. Comput. J. **53**, 443–464 (2010)
68. Sibson, R.: SLINK: an optimally efficient algorithm for the single link cluster method. Comput. J. **16**, 30–34 (1973)
69. Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge (1996)
70. Steinley, D.: K-means clustering: a half-century synthesis. Br. J. Math. Stat. Psychol. **59**, 1–34 (2006)
71. Steinley, D., Brusco, M.J.: Initializing K-means batch clustering: a critical evaluation of several techniques. J. Classif. **24**, 99–121 (2007)
72. Van Mechelen, I., Bock, H.-H., De Boeck, P.: Two-mode clustering methods: a structured overview. Stat. Methods Med. Res. **13**, 363–394 (2004)
73. van Rijsbergen, C.J.: The Geometry of Information Retrieval. Cambridge University Press, Cambridge (2004)
74. Van Rooij, A.C.M.: Non-Archimedean Functional Analysis. Dekker, New York (1978)
75. Vempala, S.S.: The Random Projection Method. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 65. American Mathematical Society, Providence (2004)
76. Volovich, I.V.: Number theory as the ultimate physical theory. Technical report (1987) Preprint No. TH 4781/87, CERN, Geneva

77. Volovich, I.V.: p-Adic string. Class. Quantum Gravity **4**, L83–L87 (1987)
78. Volovich, I.V.: Number theory as the ultimate physical theory. p-Adic Numbers Ultrametric Anal. Appl. **2**, 77–87 (2010)
79. Weyl, H.: Symmetry. Princeton University Press, Princeton (1983)
80. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**, 645–678 (2005)

A New Algorithm for Inferring Hybridization Events Based on the Detection of Horizontal Gene Transfers

Vladimir Makarenkov, Alix Boc, and Pierre Legendre

Abstract Hybridization and horizontal gene transfer are two major mechanisms of reticulate evolution. Both of them allow for a creation of new species by recombining genes or chromosomes of the existing organisms. An effective detection of hybridization events and estimation of their evolutionary significance have been recognized as main hurdles of the modern computational biology. In this article, we underline common features characterizing horizontal gene transfer and hybridization phenomena and describe a new algorithm for the inference and validation of the diploid hybridization events, when the newly created hybrid has the same number of chromosomes as the parent species. A simulation study was carried out to examine the ability of the proposed algorithm to infer correct hybrids and their parents in various practical situations.

Keywords Additive tree • Phylogenetic tree • Horizontal gene transfer • Hybridization

1 Introduction

Horizontal gene transfer (HGT) and hybridization, which are often followed by genetic or chromosomic recombination, have been recognized as major forces contributing to the formation of new species. Both of these evolutionary mechanisms are important parts of the reticulate evolution phenomenon which requires a

V. Makarenkov (✉)
Département d'Informatique, Université du Québec à Montréal, C.P. 8888, succursale Centre Ville, Montréal, QC, Canada H3C 3P8
e-mail: makarenkov.vladimir@uqam.ca

A. Boc • P. Legendre
Université de Montréal, C.P. 6128, succursale Centre-ville Montréal, QC, Canada H3C 3J7
e-mail: alix.boc@umontreal.ca; pierre.legendre@umontreal.ca

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_17,

network topology for its correct graphical representation. Phylogenetic networks are a generalization of phylogenetic (or additive) trees which have been systematically used in biological and bioinformatics studies since the publication of Darwin's *On the Origin of Species by Means of Natural Selection* [10] in order to represent the process of species evolution. Phylogenetic trees and networks are usually reconstructed according to similarities and differences between genetic or morphological characteristics of the observed species (i.e., taxa or objects). The tree reconstruction can rely either on distance-based methods [35] or on character-based methods [17]. When distance-based methods are considered, the tree building process is usually twofold: the distances are first estimated from character data and a tree is then inferred from the distance estimates. The character-based methods assume that genetic sequences evolve from a common ancestor by a process of mutation and selection without mixing (e.g., without HGT or hybridization events).

However, phylogenetic trees cannot be used to represent complex reticulate evolutionary mechanisms such as hybridization, HGT, recombination, or gene duplication followed by gene loss. Phylogenetic networks have become the models of choice when reticulation events have influenced species evolution [18, 19]. One example of phylogenetic networks is a reticulogram, i.e. reticulated cladogram, which is an undirected connected graph capable of retracing reticulate evolutionary patterns existing among the given organisms [23]. Since their introduction in 2002, reticulograms have been used to portray a variety of phylogenetic and biogeographic mechanisms, including hybridization, microevolution of local populations within a species, and historical biogeography of dispersion events [23, 26].

HGT, which is also called lateral gene transfer, is one of the main mechanisms contributing to the diversification of microbial genomes. HGT consists of a direct transfer of genetic material from one lineage to another. Bacteria and viruses have developed complex mechanisms of the acquisition of new genes by HGT to better adapt to changing environmental conditions [11, 41]. Two main HGT detection approaches exist in the literature. First of them proceeds by sequence analysis of the host genome in order to identify the genomic fragments with atypical GC content or codon usage patterns [22]. The second approach compares a morphology-based species tree, or a molecular tree inferred from a molecule which is supposed to be unaffected by HGT (e.g., 16S rRNA), with a phylogeny of the considered gene. When bacterial or viral data are examined, the observed topological differences between two trees can be often explained by HGT. The second approach comprises numerous heuristic algorithms, including the network-based models introduced by Hein [15], von Haeseler and Churchill [38], and Page and Charleston [9, 31, 32]. Mirkin et al. [28] described a tree reconciliation method for integrating different gene trees into a unique species phylogeny. Maddison [25] and then Page and Charleston [32] were first to present the set of evolutionary constraints that should be satisfied when inferring HGT events. Several recently proposed methods deal with the approximation of the Subtree Prune and Regraft (SPR) distance which is used to estimate the minimum possible number of HGTs. Bordewich and Semple [8] showed that computing the SPR distance between two rooted binary trees is NP-hard. A model allowing for mapping several gene trees into a species tree was

introduced by Hallett and Lagergren ([14], LatTrans algorithm). Mirkin et al. [29] described an algorithm for the reconciliation of phyletic patterns with a species tree by simultaneously considering gene loss, gene emergence, and gene transfer events. Mirkin et al. [29] showed that in each situation their algorithm, which can be seen as one of the main references in this field, provided a parsimonious evolutionary scenario for mapping gene loss and gain events into a species phylogenetic tree. Nakhleh et al. [30] and Than and Nakhleh [37] put forward the RIATA-HGT heuristic based on the divide-and-conquer approach. Boc et al. [6] introduced a new HGT inference algorithm, HGT-Detection, and showed that it is considerably faster than the exhaustive HGT detection strategy implemented in LatTrans, while being identical in terms of accuracy. HGT-Detection was also proved to be faster and generally more reliable than RIATA-HGT. The HGT-Detection algorithm will be considered as a backbone procedure for the hybrid detection technique that we introduce in this article.

Hybridization is another major process of reticulate evolution [2]. It is very common among plants, fish, amphibians, and reptiles and is rather rare among other groups of species, including birds, mammals, and most arthropods [27]. The new species is created by the process of recombination of genomes of different parent species. When the new species has the same number of chromosomes as its parents, the process is called *diploid hybridization*. When the new species has the sum of the number of the parent's chromosomes, the process is called *polyploid hybridization*. In this study, we will assume that new species has been created by the process of diploid hybridization. Most of the hypotheses and conclusions about hybridization rely on morphological data, and in many situations, these hypotheses have not been rigorously tested by simulations [20]. The majority of the works addressing the issue of the hybrids detection aim at calculating the minimal number of hybridization events that are necessary to reconcile the given tree topologies [3, 8]. Some of them proceed by estimating the SPR distance between a pair of rooted trees [1, 39, 40]. The main drawback of these methods is that most of them can deal only with a small number of hybrids and none of them offers the possibility of a statistical validation of the obtained hybridization events.

In this article, we propose a new algorithm for inferring a minimum number of statistically validated hybridization events that are necessary to reconcile the set of gene trees belonging to different parents (i.e., male and female gene trees) under the hypothesis of diploid hybridization. The new method will use the common features characterizing HGT and hybridization processes by separating the task of detecting hybridization events into several sub-tasks, each of which could be tackled by solving an equivalent HGT detection problem. A statistical validation procedure allowing one to assess the bootstrap support of the proposed hybrids and their parents will be incorporated into the new algorithm. A simulation study along with an application example will also be presented in the article.

2 Definitions and Basic Concepts

This section recalls some basic definitions concerning phylogenetic trees and tree metrics following the terminology of Barthélemy and Guénoche [4]. The distance $\delta(x,y)$ between two vertices x and y in a phylogenetic tree T is defined as the sum of the edge lengths of the unique path connecting x and y in T. Such a path is denoted (x,y). A leaf is a vertex of degree one.

Definition 1. Let X be a finite set of n taxa. A dissimilarity d on X is a nonnegative function on $(X \times X)$ such that for any x, y from X:

(1) $d(x, y) = d(y, x)$, and
(2) $d(x, y) = d(y, x) \geq d(x, x) = 0$.

Definition 2. A dissimilarity d on X satisfies the four-point condition if for any x, y, z, and w from X:

$$d(x, y) + d(z, w) \leq \text{Max}\{d(x, z) + d(y, w); d(x, w) + d(y, z)\}. \quad (1)$$

Definition 3. For a finite set X, a phylogenetic tree (i.e., an additive tree or an X-tree) is an ordered pair (T, φ) consisting of a tree T, with vertex set V, and a map φ: $X \rightarrow V$ with the property that, for all $x \in X$ with degree at most two, $x \in \varphi(X)$. A phylogenetic tree is called binary if φ is a bijection from X into the leaf set of T and every interior vertex has degree three. The main theorem linking the four-point condition and phylogenetic trees (i.e., phylogenies) is as follows:

Theorem 1 (Zarestskii, Buneman, Patrinos, Hakimi, and Dobson).
Any dissimilarity satisfying the four-point condition can be represented by a phylogenetic tree T such that for any x, y from X, $d(x, y)$ is equal to the length of the path linking the leaves x and y in T. This dissimilarity is called a tree metric. Furthermore, this tree is unique.

Figure 1 presents an example of a tree metric on the set X of five taxa and the corresponding phylogenetic tree. Note that raw biological data rarely give rise directly to a tree metric (i.e., to a phylogenetic tree) but rather to a dissimilarity not satisfying the four-point condition. Biologists have to infer tree metrics and the corresponding trees by fitting the given dissimilarity with a tree metric according to a specific criterion.

	x2	x3	x4	x5
x1	6	5	4	6
x2		5	6	4
x3			5	5
x4				6

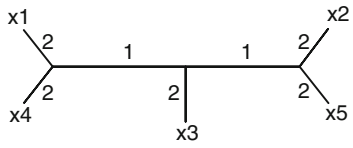


Fig. 1 A tree metric on the set X of five taxa and the associated phylogenetic tree with five leaves

3 HGT Detection Problem and Related Optimization Criteria

The problem of finding the minimum number of HGTs that are necessary to transform one phylogenetic tree into another (i.e., also known as *Subtree Transfer Problem*) has been shown to be NP-hard [16]. Here we recall the main features of the HGT-Detection algorithm [6] intended for inferring HGT events. This algorithm proceeds by a progressive reconciliation of the given species and gene phylogenetic trees, denoted T and T', respectively. Usually, the species tree T is inferred from the gene that is refractory to HGT and genetic recombination. This tree represents the direct, or tree-like, evolution. The gene tree T' represents the evolution of the given gene which is supposed to undergo horizontal transfers.

At each step of the algorithm, all pairs of edges of the species tree T are tested against the hypothesis that a HGT has occurred between them. Thus, the original species phylogenetic tree T is progressively transformed into the gene phylogenetic tree T' via a series of SPR moves (i.e., gene transfers). The topology of the gene tree T' is fixed throughout the transformation process. The goal of the method is to find the minimum possible sequence of trees $T, T_1, T_2, \ldots, T'$ transforming T into T'. Obviously, a number of necessary biological rules should be taken into account. For example, the transfers within the same lineage should be prohibited [14,25,32]. The subtree constraint we consider here (see Appendix) allows us to take into account all necessary evolutionary rules.

We will consider the four following optimization criteria which can be used to select optimal transfers at each step of the algorithm: least-squares, the Robinson and Foulds topological distance, the quartet distance, and the bipartition dissimilarity. The first employed optimization criterion is the *least-squares function* Q. It is defined as follows:

$$Q = \sum_i \sum_j (d(i,j) - \delta(i,j))^2, \tag{2}$$

where $d(i,j)$ is the distance between the leaves i and j in the species tree T at the first step of the algorithm (or the transformed species trees at the following steps of the algorithm) and $\delta(i,j)$ is the distance between the leaves i and j in the gene tree T'.

The second criterion we use in the transfer detection part of our algorithm is the *Robinson and Foulds* (RF) *topological distance*. The RF metric [33] is an important and frequently used tool for comparing phylogenetic trees. This distance is equal to the minimum number of elementary operations, consisting of merging and splitting nodes, which are necessary to transform one tree into the other. This distance is twice the number of bipartitions present in one of the trees and absent in the other. When the RF distance is considered, we use it as the optimization criterion in the following way: all possible transformations of the species tree, consisting of SPR moves of its subtrees satisfying the biological constraints, are evaluated in such a

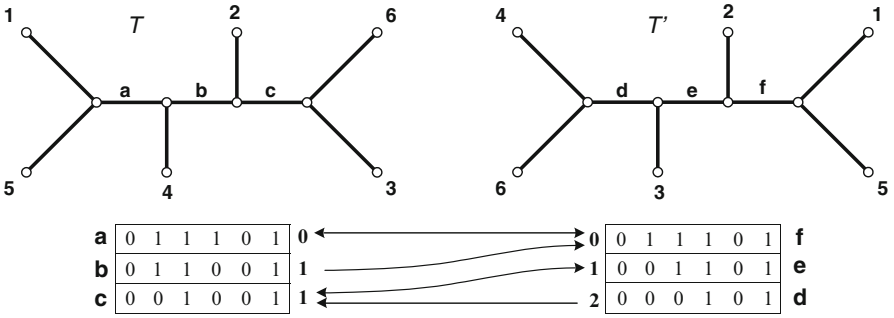


Fig. 2 Trees T and T' and their bipartition tables. *Each row* of the bipartition table corresponds to an internal edge of the tree. *Arrows* indicate the correspondence between the bipartition vectors in the two tables. *Value in bold* near each vector indicates the corresponding distance

way that the RF distance between the transformed species tree T_1 and the gene tree T' is computed. The subtree transfer yielding the minimum of the RF distance between T_1 and T' is then selected.

The third considered criterion is the *quartet distance* (QD). QD is the number of quartets, or subtrees induced by four leaves, which differ between the compared trees. We can use this criterion in the same way that the RF metric.

The fourth optimization criterion is the *bipartition dissimilarity* (BD), first defined in Boc et al. [6]. Assume that T and T' are binary phylogenetic trees on the same set of leaves. A bipartition vector (i.e., split or bipartition) of the tree T is a binary vector induced by an internal edge of T. Let **BT** be the bipartition table of the internal edges of T and **BT**′ be the bipartition table of the internal edges of T'. The bipartition dissimilarity bd between T and T' is defined as follows:

$$bd = \left(\sum_{a \in \mathrm{BT}} \underset{b \in \mathrm{BT}'}{\mathrm{Min}} (\mathrm{Min}(d(a,b); d(a,\bar{b}))) + \sum_{b \in \mathrm{BT}'} \underset{a \in \mathrm{BT}}{\mathrm{Min}} (\mathrm{Min}(d(b,a); d(b,\bar{a}))) \right) / 2, \tag{3}$$

where $d(a,b)$ is the Hamming distance between the bipartition vectors a and b ($\bar{a}$ and $\bar{b}$ are the complements of a and b, respectively). The bipartition dissimilarity can be seen as a refinement of the RF metric which takes into account only the identical bipartitions. For example, the bipartition dissimilarity between the trees T and T' with six leaves presented in Fig. 2 is computed as follows: $bd(T,T) = ((0+1+1)+(0+1+2))/2 = 2.5$.

In our simulation study described below we presented the results obtained using the bipartition dissimilarity as the optimization criterion because it provided the best overall simulation performances compared to the RF and QD distances and least-squares.

4 Algorithm Description

In this section we describe the main features of the new algorithm for detecting hybridization events. The statistical bootstrap validation will be performed for each hybrid species and only the hybrids with a significant bootstrap support will be included in the final solution. The new algorithm for identifying hybridization events proceeds by a reconciliation of the given pairs of gene trees, constructed for genes inherited from different parents. A modified version of the procedure for detecting HGTs described in Boc et al. [6] will be integrated in our new algorithm. Let $\mathbf{G}_{\mathrm{m}}$ be the set of genes that can be inherited from a male parent only and $\mathbf{G}_{\mathrm{f}}$ be the set of genes that can be inherited from a female parent only. In practice, nuclear and chloroplast genes often play the roles of $\mathbf{G}_{\mathrm{m}}$ and $\mathbf{G}_{\mathrm{f}}$, respectively. We assume that for each given gene there exists a set of orthologous gene sequences (i.e., sequences that originated from a single gene of the last common ancestor) that can be used to build a phylogenetic gene tree. Each gene is thus originally represented by a multiple sequence alignment of amino acids or nucleotides.

Step 1. For the multiple sequence alignments characterizing the male genes in $\mathbf{G}_{\mathrm{m}}$ we infer a set of phylogenetic male gene trees $\mathbf{T}_{\mathrm{m}}$ and for the multiple sequence alignments characterizing the female genes in $\mathbf{G}_{\mathrm{f}}$ we infer a set of phylogenetic female gene trees $\mathbf{T}_{\mathrm{f}}$; one gene tree by alignment is reconstructed. The trees can be inferred using methods such as Neighbor-Joining [34], PhyML [13], RaxML [36], or one of the phylogenetic inference algorithms from the PHYLIP package [12]. We then root all the trees in $\mathbf{T}_{\mathrm{m}}$ and $\mathbf{T}_{\mathrm{f}}$ according to biological evidence or using the outgroup or midpoint strategy and select the optimization criterion, which can be least-squares, the Robinson and Foulds topological distance [33], the quartet distance, or the bipartition dissimilarity [6].

Step 2. For each pair of gene trees T and T', such that $T \in \mathbf{T}_{\mathrm{m}}$ and $T' \in \mathbf{T}_{\mathrm{f}}$, we use the HGT-Detection algorithm [6] to identify first HGTs that are required to transform T into T'. The HGT-Detection program carries out a fast and accurate heuristic algorithm for computing a minimum-cost SPR transformation of the given (species) tree T into the given (gene) tree T'. Figure 3 shows how a species tree is transformed into a gene tree by applying a transfer (SPR move) between its subtrees (i.e., edges adjacent to the species C and E). After this SPR move, T and T' have the identical topology. Second, we repeat the procedure by inversing the roles of T and T'. Now we look for HGTs that are necessary to transform T' into T. The statistical bootstrap support of each obtained transfer is then assessed as defined in Boc et al. [6] and Boc and Makarenkov [5]. We identify as potential hybrids the species that receive transfers from different parents in T and T' (e.g., species H in Fig. 4 which receives a transfer from the species C and B; here, C and B can be viewed as the parents of H).

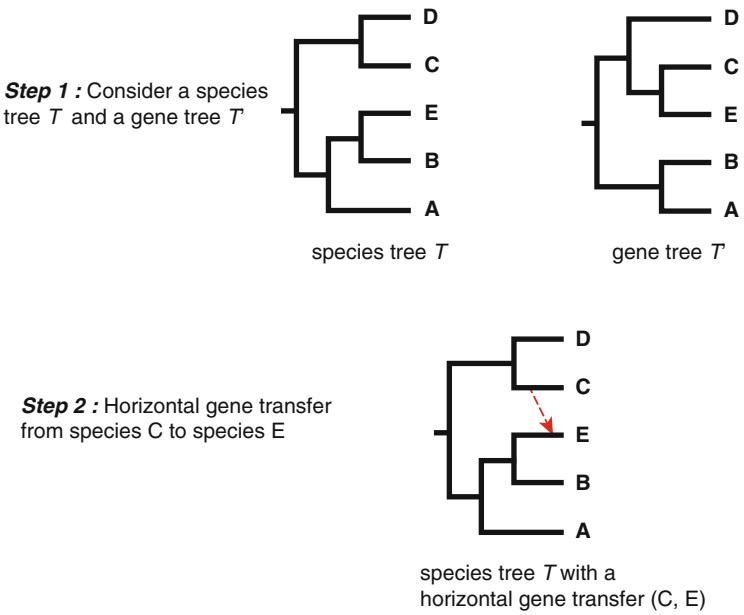


Fig. 3 Horizontal gene transfer (i.e., SPR move) from species C to species E is necessary to transform the topology of the species tree T into the topology of the gene tree T'

Final Step. All the obtained horizontal transfers are classified according to their statistical support to establish a ranked list of predicted hybrid species and their parents. In our algorithm, a confirmed hybrid species is a species that receives a transfer stemming from different parents in at least two gene trees (such that at least one of them is from $\mathbf{T}_{\mathrm{m}}$ and at least one of them is from $\mathbf{T}_{\mathrm{f}}$) with a fixed minimum confidence score (i.e., average bootstrap support). When multiple trees from $\mathbf{T}_{\mathrm{m}}$ and $\mathbf{T}_{\mathrm{f}}$ are involved, this score is computed as the mean value of the average bootstrap scores found for the two groups of parents. If the gene trees are considered without uncertainties (i.e., no bootstrap validation performed), then all hybrid species found by the algorithm can be included in the final solution. The main steps of the new algorithm are presented below (see Algorithm 1). Its time complexity is the following:

$$O(m \times f \times r \times (C(\mathrm{TreeInf}) + n^4)), \tag{4}$$

where m and f are the cardinalities of the sets $\mathbf{G}_{\mathrm{m}}$ and $\mathbf{G}_{\mathrm{f}}$, respectively, r is the number of replicates in bootstrapping, $C(\mathrm{TreeInf})$ is the time complexity of the tree inferring method used to infer trees from the gene sequences, and n^4 is the time complexity of the HGT-Detection algorithm [6] applied to the given pair of species

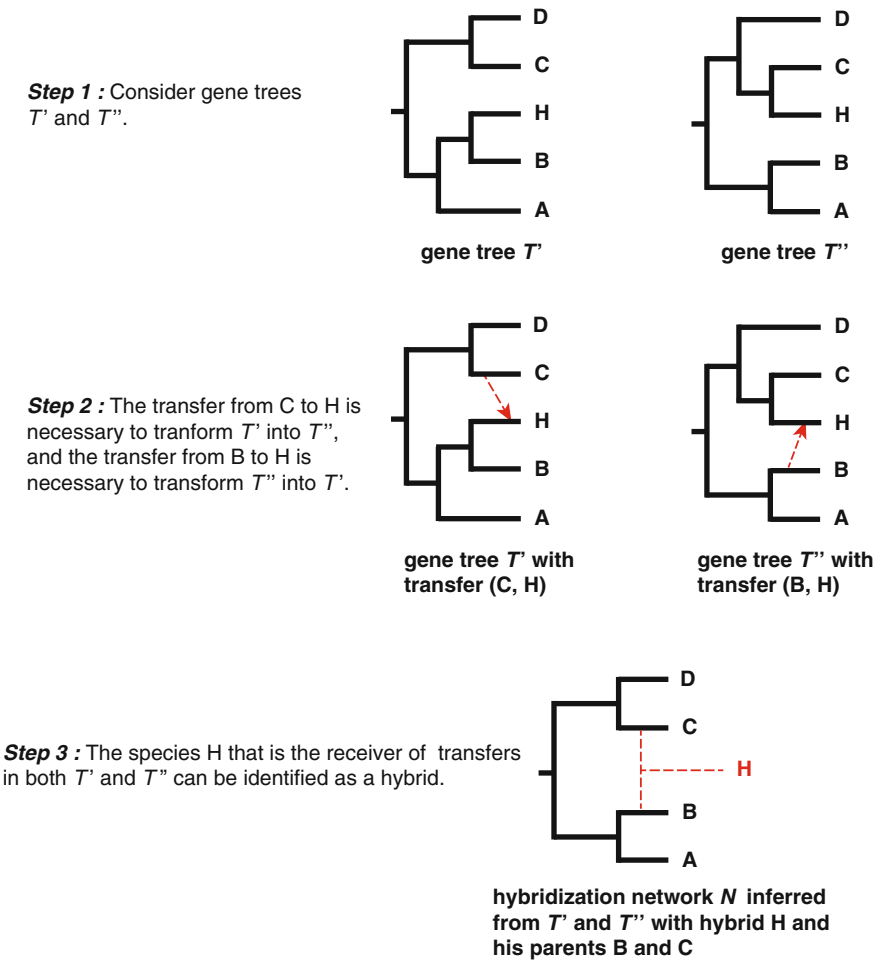


Fig. 4 The main idea of the hybrid inference method: the species H that receives transfers (Step 2) in both gene trees T' and T'' is identified as a hybrid. The hybridization network N is thus obtained (Step 3)

and gene trees with n leaves. Given that the time complexity of the PhyML [13] method which we used in our simulation study is $O(pnl)$, where p is the number of refinement steps being performed, n is the number of species and l is the sequence length, the exact time complexity of our implementation is the following:

$$O(m \times f \times r \times n \times (p \times l + n^3)). \tag{5}$$

Algorithm 1 Main steps of the hybrids detection algorithm. See Appendix for the definition of the subtree constraint which allows one to take into account all necessary biological rules and for Theorems 2 and 3 which allow one to select optimal transfers in different practical situations

Infer all gene trees $\mathbf{T}_m$ for the set of the male genes $\mathbf{G}_m$ and all gene trees $\mathbf{T}_f$ for the set of the female genes $\mathbf{G}_f$;
Root all the trees in $\mathbf{T}_m$ and $\mathbf{T}_f$ according to biological evidence or using the outgroup or midpoint strategy;
Select the optimization criterion $OC = Q$ (least-squares), or RF (Robinson and Foulds distance), or QD (quartet distance), or BD (bipartition dissimilarity);

for each tree T from the set of the male gene trees $\mathbf{T}_m$ **do**
  for each tree T' from the set of the female gene trees $\mathbf{T}_f$ **do**
    if there exist identical subtrees with two or more leaves in T and T' **then**
      Decrease the size of the problem by collapsing them in both T and T';
    end if
    Compute the initial value of OC between T_0 and T';
    (*) $T_0 = T$; // or $T_0 = T$' - when repeated
    $k = 1$; // k is the step index

    while $OC \neq 0$ **do**
      Find the set of all eligible horizontal transfers (i.e., SPR moves) at step k (denoted as EHT_k);
      The set EHT_k contains only the transfers satisfying the subtree constraint;
      while transfers satisfying the conditions of Theorems 3 and 2 exist **do**
        if there exist transfers $\in EHT_k$ and satisfying the conditions of Theorem 3 **then**
          Carry out the SPR moves corresponding to these transfers;
        end if
        if there exist transfers $\in EHT_k$ and satisfying the conditions of Theorem 2 **then**
          Carry out the SPR moves corresponding to these transfers;
        end if
      end while
      Carry out all remaining SPR moves corresponding to transfers satisfying the subtree constraint;
      Compute the value of OC to identify the direction of each transfer;
      $k = k + 1$;
      Collapse the same subtrees in T_k and T'; // or in T_k and T - when repeated
      Compute the value of OC between T_k and T'; // or between T_k and T - when repeated
    end while

    Repeat the procedure above by inversing the roles of T and T', starting from (*);
    Identify species (potential hybrids) such that they receive transfers from different species in T and T';
  end for
end for

Classify all horizontal transfers and potential hybrids found;
Repeat the procedure above twice using the replicates of T and T' (obtained from the replicates of the multiple sequence alignments corresponding to T and T') to establish the list of predicted hybrid species and their parents with their bootstrap support.

5 Simulation Study

A Monte Carlo study was conducted to test the capacity of the new algorithm to identify correct hybrid species. Gene trees were assumed not to contain uncertainties and thus the simulations were carried out with tree-like data only (i.e., sequence data were not involved). We examined how the new algorithm performs depending on the number of observed species, the rate of hybridization, and the number of hybrid species artificially added. The measured hybridization rate is the ratio of genes originating from the different parents (i.e., male and female species).

First, a binary gene tree T was generated using the random tree generation procedure described in [21]. An improved version of this procedure was included in our T-Rex package [7]. As we did not consider sequence data in these simulations, the edge lengths of the trees were not taken into account here. In each experiment, we considered ten replicates of the gene tree T, assuming that some of them originated from the male and some of them from the female parent species.

Second, for a fixed hybridization rate h (h varied from 1 to 5 in our simulations) we randomly selected in the first h replicates of T the same species (or group of species) as Parent P_1 and in the remaining (10-h) replicates of T another species (or group of species) as Parent P_2. Obviously, when the groups were considered, all the species in P_1 were different from the species in P_2. A new edge with the hybrid species H was then added to each of the first h gene trees. It was connected to the edge separating P_1 from the rest of the tree. Similarly, the edge with the same hybrid species H was added to each of the remaining (10-h) gene trees, and connected each time to the edge separating P_2 from the rest of the tree. This step was repeated sh times, where sh denotes the number of integrated hybrid species. In our simulations, sh varied from 1 to 10.

Third, we carried out the introduced hybrid detection algorithm having as input ten replicates of the gene tree T with the hybrids added as discussed above. As the gene trees were considered without uncertainties, all transfers detected in the process were considered as relevant and were taken into account in the final solution. The bipartition dissimilarity [6] was used as the optimization criterion in the HGT-Detection procedure. The results illustrated in Figs. 5 and 6 were obtained from simulations carried out with random binary phylogenetic trees with 8, 16, 32, and 64 leaves. For each tree size (8–64), each number of hybrid species (1–10) and each hybridization rate (10–50 %), 1,000 replicated datasets were generated.

The true detection rate (i.e., true positives) was measured as a percentage of the correctly recovered hybrid species that were generated. The performances of the new algorithm are more noticeable for large trees (see Figs. 5 and 6, cases c–d) and a small number of hybrids. The quality of the obtained results decreases when the number of species decreases. For instance, to detect 10 hybrids in trees with 8 possible parental species seems to be a very tricky task, especially when the hybridization rate varies from 10 to 30 % (i.e., $h = 1, 2$ and 3; see Figs. 5

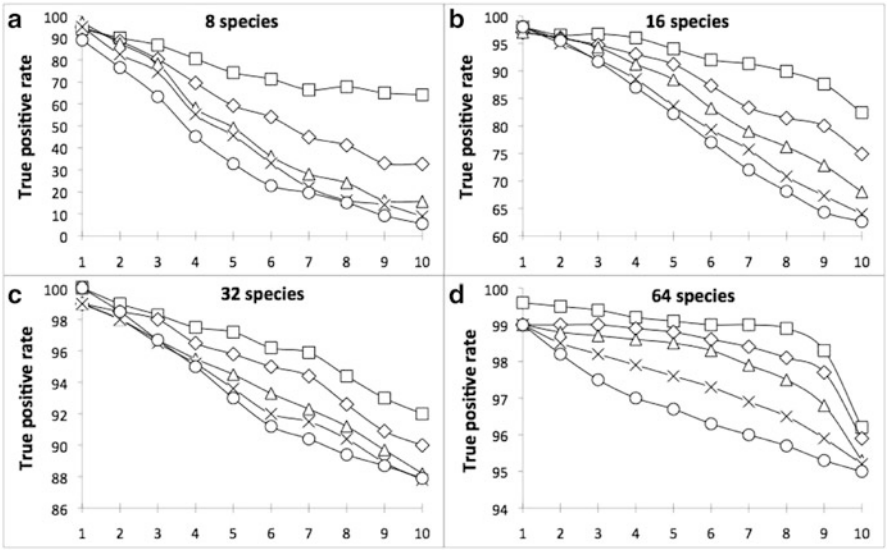


Fig. 5 Average true positive hybrid detection rate obtained for binary trees with 8 (**a**), 16 (**b**), 32 (**c**), and 64 (**d**) leaves. The five presented curves correspond to the hybridization rate *h* of 50 % (*open square*), 40 % (*open diamond*), 30 % (*open triangle*), 20 % (*times symbol*), and 10 % (*open circle*). The abscissa axis reports the number of hybrid species. Each presented value is an average computed over 1,000 replicates

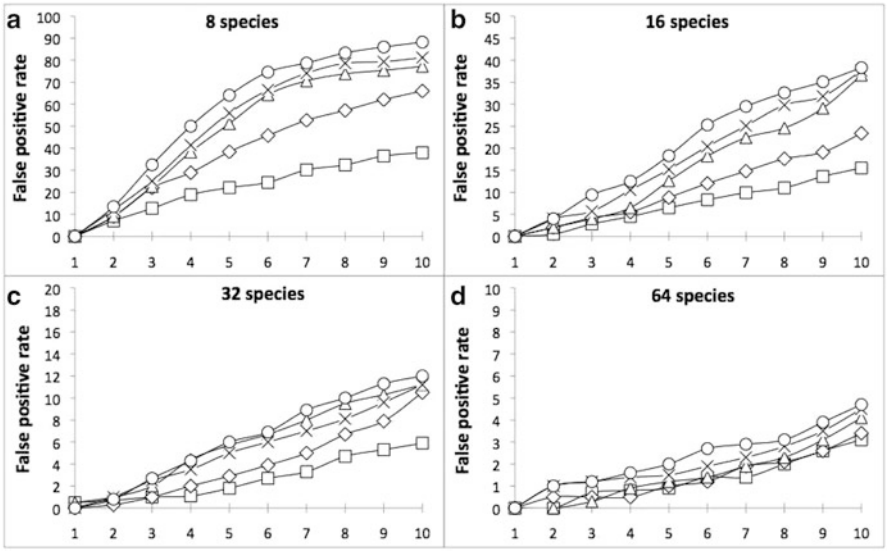


Fig. 6 Average false positive hybrid detection rate obtained for binary trees with 8 (**a**), 16 (**b**), 32 (**c**), and 64 (**d**) leaves. The five presented curves correspond to the hybridization rate *h* of 50 % (*open square*), 40 % (*open diamond*), 30 % (*open triangle*), 20 % (*times symbol*), and 10 % (*open circle*). The abscissa axis reports the number of hybrid species. Each presented value is an average computed over 1,000 replicates

and 6, case a). Another general trend that could be noticed is that the number of true positives increases and the number of false positives decreases as the hybridization rate grows (i.e., the best results were always observed for $h = 4$ and 5).

6 Application Example

6.1 *Detecting Hybrid Species in the New Zealand's Alpine Ranunculus Dataset*

We studied the evolution of 6 different genes belonging to 14 organisms of the alpine *Ranunculus* plants originally described in Lockhart et al. [24], and then analyzed in Joly et al. [20]. The latter authors presented a novel parametric approach for statistically distinguishing hybridization from incomplete lineage sorting based on minimum genetic distances of nonrecombining genes. Joly and colleagues applied their method to detect hybrids among the New Zealand's alpine buttercups (*Ranunculus*). Fourteen individuals of *Ranunculus* belonging to six well-defined species were sequenced in five chloroplast regions (*trnC-trnD*, *trnL-trnF*, *psbA-trnH*, *trnD-trnT*, and *rpL16*). Those sequences were concatenated in the analysis conducted by Joly et al. [20]. In this study, they will be analyzed separately using our new algorithm. Note that in most flowering plants, chloroplast genes are inherited by hybrids from the female parent only. In contrast, the sequences from another considered gene, the internal transcribed spacer (*nrITS*) region, were assumed to be inherited from the male parent only.

We first reconstructed from the original sequences the topology of the $nrITS$ gene tree (Fig. 7) as well as those of the *psbA*, *rpL16*, *trnC*, *trnD*, and *trnL* gene trees (Fig. 8).

The hybrid species detection was performed by the new algorithm and five possible hybrid species were identified (see Table 1) along with their parents and the corresponding bootstrap scores. All transfers found, when gradually reconciling the *nrITS* gene tree with the *psbA*, *rpL16*, *trnC*, *trnD*, and *trnL* gene trees, are illustrated in Fig. 9. As a backbone tree topology here we used the species tree built with respect to the species chronogram of the alpine *Ranunculus* presented in [20, Fig. 5]. The most significant hybrid species we found was the *R. insignis* Mt Hutt. The species *R. crithmifolius* Ben Ohau and *R. crithmifolius* Mt Lyndon were identified as its parents with the bootstrap scores of 76 and 75 %, respectively. Thus, the bootstrap support of this hybrid, computed as the average of its parents bootstrap scores, is equal to 75.5 %.

Our algorithm also suggested multiple hypotheses for an eventual hybrid species *R. crithmifolius* Mt Lyndon. The first hypothesis assumes that its parents could be *R. crithmifolius* Castle Hill (47 %) and *R. insignis* Mt Hutt (84 %), combining for

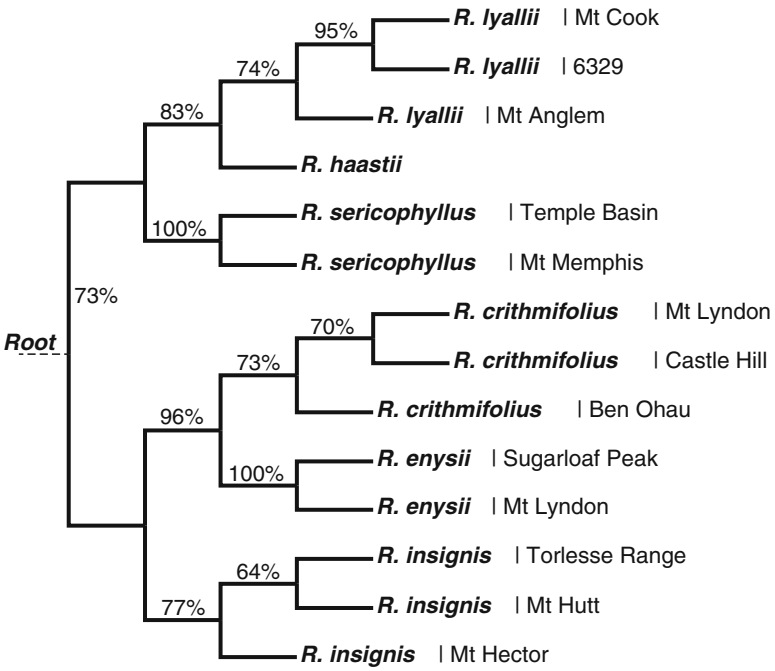


Fig. 7 Phylogenetic tree of the gene $nrITS$ built for 14 organisms of alpine *Ranunculus* using the PhyML method [13]. The bootstrap scores of the internal edges of the tree are indicated

the average bootstrap support of 65.5 %. The second hypothesis suggests that its parents could be the ancestor of *R. haastii*, *R. lyallii 6329*, *R. lyallii* Mt Anglem, *R. lyallii* Mt Cook, *R. sericophyllus* Mt Memphis, and *R. sericophyllus* Temple Basin as the first parent, with the bootstrap of 45.5 %, and *R. insignis* Mt Hutt as the second parent, with the bootstrap support of 84 %, providing the average support of 64.5 %. The third hypothesis concerning *R. crithmifolius* Mt Lyndon states that the parents of this organism could be in fact the ancestor of *R. haastii*, *R. lyallii 6329*, *R. lyallii* Mt Anglem, *R. lyallii* Mt Cook, *R. sericophyllus* Mt Memphis, and *R. sericophyllus* Temple Basin, with the bootstrap score of 45.5 %, and the species *R. crithmifolius* Castle Hill (47 %), giving the average bootstrap support of 46 %. As discussed in [20], hybridization is a likely hypothesis for the chloroplast lineage present in *R. crithmifolius* from Mt Lyndon and *R. insignis* from Mt Hutt. Our analysis supported both these hypotheses while suggesting an additional hybrid possibility in this dataset, concerning *R. insignis* Torlesse Range (see Table 1). The latter species was also identified as a potential hybrid with the bootstrap support of 52.5 %, whereas *R. insignis* Mt Hutt (58 %) and *R. enysii* Sugarloaf Peak (47 %) were categorized as its parents.

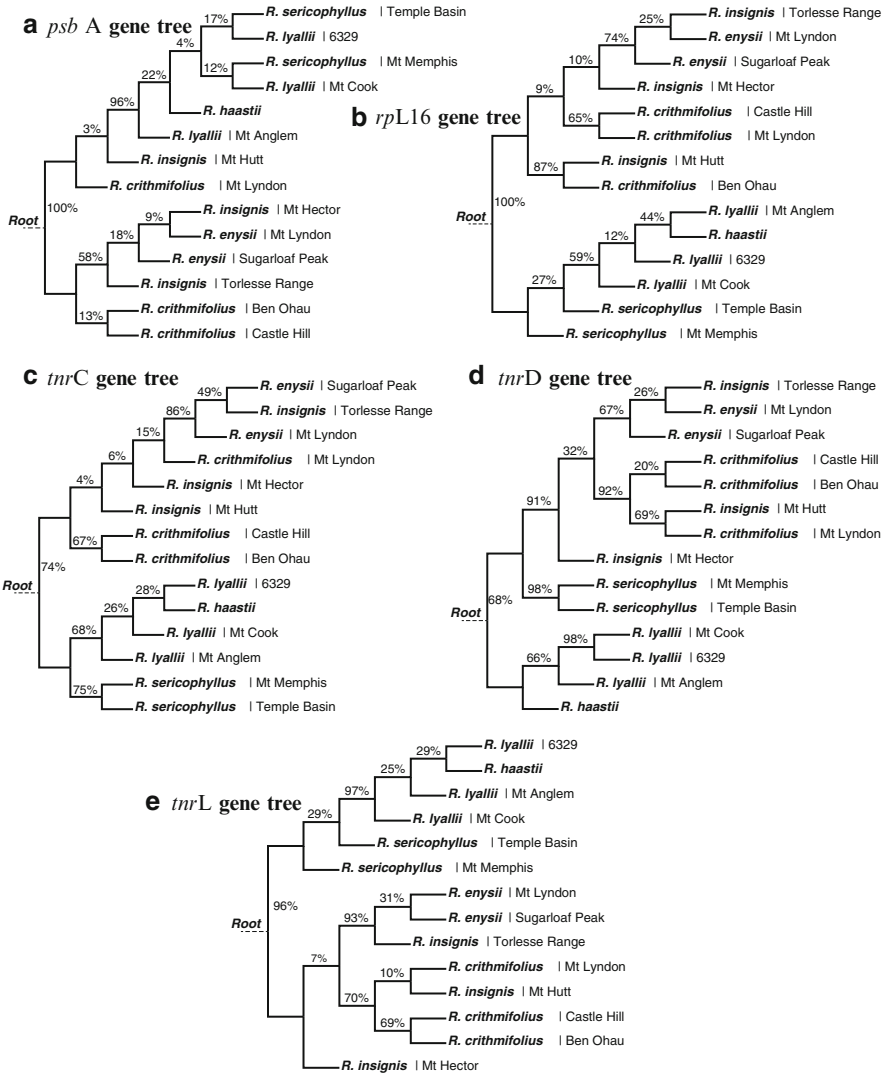


Fig. 8 Phylogenetic trees of the genes *psbA*, *rpL16*, *trnC*, *trnD*, and *trnL* built for 14 organisms of alpine *Ranunculus* using the PhyML method [13]. The bootstrap scores of the internal edges of the tree are indicated

7 Conclusion

We described a new algorithm for detecting and validating diploid hybridization events and thus for identifying the origins of hybrid species. To the best of our knowledge no algorithms including a statistical validation of the retraced hybrids and their parents by bootstrap analysis have been proposed in the literature.

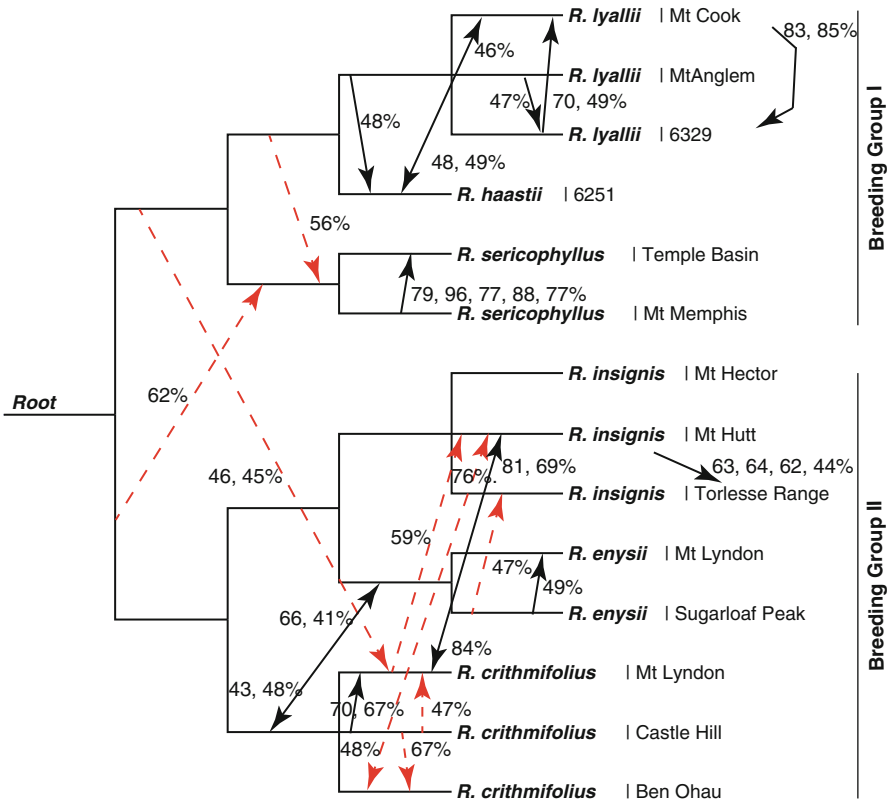


Fig. 9 Species tree for the 14 considered *Ranunculus* organisms with horizontal transfers mapped into it. *Dashed arrows* depict the transfers stemming from the gene *nrITS*. *Full arrows* depict the transfers stemming from the genes *psbA*, *rpL16*, *trnC*, *trnD*, and *trnL*. A potential hybrid species should be a receiver of at least one *dashed arrow* and at least one *full arrow* originating from different sources

We showed that the problem of detecting HGTs can be viewed as a sub-problem of a hybrid detection problem when multiple male and female genes are considered. The introduced algorithm subdivides the multi-gene reconciliation problem on several sub-problems searching for optimal scenarios of SPR moves that are required to reconcile gene trees associated with genes originating from different parents (male or female species). To find such optimal tree reconciliation scenarios, we use a specific version the HGT-Detection [6] algorithm, which is a fast and accurate heuristic for inferring HGT events. Our simulation study suggests that the best detection results are constantly obtained with large trees and a small number of hybrids. Regarding the optimization criterion, the bipartition dissimilarity usually provided better results compared to the classical criteria, such as the Robinson and Foulds distance, the quartet distance, and least-squares. As a future development, it would be interesting to see how the hybrid detection results would change if the trees with uncertainties (i.e., trees inferred from the sequence data) are considered.

Table 1 Hypothetical hybrids of the considered alpine *Ranunculus* organisms based on the transfer scenarios presented in Fig. 9

Hybride	Parent 1	Parent 2	Average hybrid support
R. insignis Mt Hutt	*R. crithmifolius* Ben Ohau (76 %)	*R. crithmifolius* Mt Lyndon (75 %)	75.5 %
R. crithmifolius Mt Lyndon	*R. crithmifolius* Castle Hill (47 %)	*R. insignis* Mt Hutt (84 %)	65.5 %
R. crithmifolius Mt Lyndon	Ancestor of (*R. haastii* , *R. lyallii 6329* , *R. lyallii* Mt Anglem, *R. lyallii* Mt Cook, *R. sericophyllus* Mt Memphis, *R. sericophyllus* Temple Basin) (45.5 %)	*R. insignis* Mt Hutt (84 %)	64.5 %
R. insignis Torlesse Range	*R. insignis Mt Hutt (58 %)*	*R. enysii* Sugarloaf Peak (47 %)	52.5 %
R. crithmifolius Mt Lyndon	Ancestor of (*R. haastii* , *R. lyallii 6329* , *R. lyallii* Mt Anglem, *R. lyallii* Mt Cook, *R. sericophyllus* Mt Memphis, *R. sericophyllus* Temple Basin) (45 %)	*R. crithmifolius* Castle Hill (47 %)	46 %

Each row reports the hybrid, two eventual parents, and the corresponding bootstrap supports

Appendix

This appendix includes the definition of the subtree constraint (Fig. 10) used in the hybrid detection algorithm (Algorithm 1). This constraint, originally formulated in [6], allows one to take into account all evolutionary rules that should be satisfied when inferring horizontal gene transfers. This appendix also includes Theorems 2 and 3 allowing one to select optimal transfers during the execution of the hybrid detection algorithm (Algorithm 1) (see [6] for their proofs).

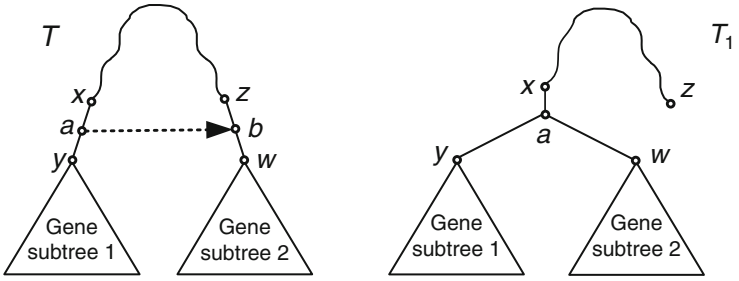


Fig. 10 Subtree constraint: the transfer between the branches (x, y) and (z, w) in the species tree T is allowed if and only if the cluster rooted by the branch (x, a), and regrouping both affected subtrees, is present in the gene tree. A single tree branch is depicted by a *plane line* and a path is depicted by a *wavy line*

Theorem 2. *If the newly formed subtree Sub_{yw} resulting from the HGT (horizontal gene transfer) is present in the gene tree T', and the bipartition vector associated with the branch (x,x_1) in the transformed species tree T_1 (Fig. 11) is present in the bipartition table of T', then the HGT from (x,y) to (z,w), transforming T into T_1, is a part of a minimum-cost HGT scenario transforming T into T' and satisfying the subtree constraint.*

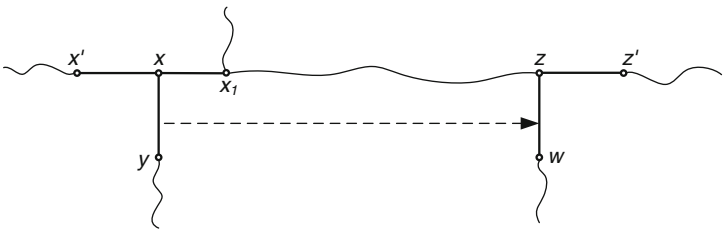


Fig. 11 HGT from the branch (x,y) to the branch (z,w) is a part of a minimum-cost HGT scenario transforming the species tree T into the gene tree T' if the bipartition corresponding to the branch (x,x_1) in the transformed species tree T_1 is present in the bipartition table of T' and the subtree Sub_{yw} is present in T'

Theorem 3. *If the newly formed subtree Sub_{yw} resulting from the HGT is present in the gene tree T', and all the bipartition vectors associated with the branches of the path (x',z') in the transformed species tree T_1 (Fig. 12) are present in the bipartition table of T', and the path (x',z') in T_1 consists of at least three branches, then the HGT from (x,y) to (z,w), transforming T into T_1, is a part of any minimum-cost HGT scenario transforming T into T' and satisfying the subtree constraint.*

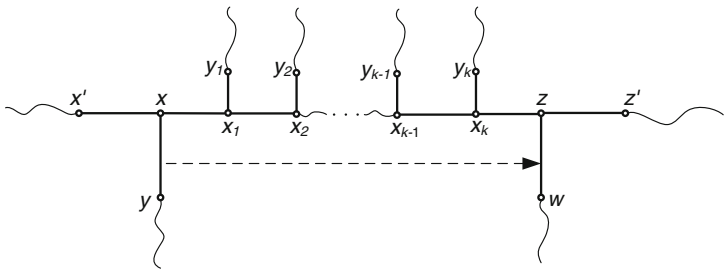


Fig. 12 HGT from the branch (x,y) to the branch (z,w) is a part of any minimum-cost HGT scenario transforming the species tree T into the gene tree T' if all the bipartitions corresponding to the branches of the path (x',z') in the transformed species tree T_1 are present in the bipartition table of T' and the subtree Sub_{yw} is present in the tree T'

References

1. Albrecht, B., Scornavacca, C., Cenci, A., Huson, D.H.: Fast computation of minimum hybridization networks. Bioinformatics **28**, 191–197 (2012)
2. Arnold, M.L.: Natural hybridization and evolution. Oxford University Press, Oxford (1997)
3. Baroni, M., Semple, C., Steel, M.: Hybrids in real time. Syst. Biol. **55**(1), 46–56 (2006)
4. Barthélemy, J.-P., Guénoche, A.: Trees and proximity representations. Wiley, New York (1991)
5. Boc, A., Makarenkov, V.: Towards an accurate identification of mosaic genes and partial horizontal gene transfers. Nucleic Acids Res. **39**, e144 (2011)
6. Boc, A., Philippe, H., Makarenkov, V.: Inferring and validating horizontal gene transfer events using bipartition dissimilarity. Syst. Biol. **59**, 195–211 (2010)
7. Boc, A., Diallo, A.B., Makarenkov, V.: T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks. Nucleic Acids Res. **40**(Web Server issue), W573–W579 (2012)
8. Bordewich, M., Semple, C.: On the computational complexity of the rooted subtree prune and regraft distance. Ann. Comb. **8**, 409–423 (2004)
9. Charleston, M.A.: Jungle: a new solution to the host/parasite phylogeny reconciliation problem. Math. Biosci. **149**, 191–223 (1998)
10. Darwin, C.: On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life, p. 502. John Murray, London (1859)
11. Doolittle, W.F.: Phylogenetic classification and the universal tree. Science **284**, 2124–2129 (1999)

12. Felsenstein, J.: PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics **5**, 164–166 (1989)
13. Guindon, S., Gascuel, O.: A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. **52**, 696–704 (2003)
14. Hallett, M., Lagergren, J.: Efficient algorithms for lateral gene transfer problems. In: El-Mabrouk, N., Lengauer, T., Sankoff, D., (eds.) Proceedings of the Fifth Annual International Conference on Research in Computational Biology, pp. 149–156. ACM, New York (2001)
15. Hein, J.: A heuristic method to reconstructing the evolution of sequences subject to recombination using parsimony. Math. Biosci. **98**, 185–200 (1990)
16. Hein, J., Jiang, T., Wang, L., Zhang, K.: On the complexity of comparing evolutionary trees. Discrete Appl. Math. **71**, 153–169 (1996)
17. Hennig, W.: Phylogenetic systematics (tr. D. Dwight Davis and Rainer Zangerl). University of Illinois Press, Urbana (1966)
18. Huson, D.H., Bryant, D.: Application of phylogenetic networks in evolutionary studies. Mol. Biol. Evol. **23**, 254–267 (2006)
19. Huson, D.H., Rupp, R., Scornavacca, C.: Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press, Cambridge (2011)
20. Joly, S., McLenachan, P.A., Lockhart, P.J.: A statistical approach for distinguishing hybridization and incomplete lineage sorting. Am. Nat. **174**, e54–e70 (2009)
21. Kuhner, M., Felsenstein, J.: A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. **11**, 459–468 (1994)
22. Lawrence, J.G., Ochman, H.: Amelioration of bacterial genomes: rates of change and exchange. J. Mol. Evol. **44**, 383–397 (1997)
23. Legendre, P., Makarenkov, V.: Reconstruction of biogeographic and evolutionary networks using reticulograms. Syst. Biol. **51**, 199–216 (2002)
24. Lockhart, P.J., McLenachan, P.A., Havell, D., Gleny, D., Huson, D., Jensen, U.: Phylogeny, radiation, and transoceanic dispersal of New Zealand alpine buttercups: molecular evidence under split decomposition. Ann. MO Bot. Gard. **88**, 458–477 (2001)
25. Maddison, W.P.: Gene trees in species trees. Syst. Biol. **46**, 523–536 (1997)
26. Makarenkov, V., Legendre, P.: From a phylogenetic tree to a reticulated network. J. Comput. Biol. **11**, 195–212 (2004)
27. Makarenkov, V., Kevorkov, D., Legendre, P.: Phylogenetic network reconstruction approaches. In: Applied Mycology and Biotechnology. International Elsevier Series, Bioinformatics, vol. 6, pp. 61–97. Elsevier, Amsterdam (2006)
28. Mirkin, B.G., Muchnik, I., Smith, T.F.: A biologically consistent model for comparing molecular phylogenies. J. Comput. Biol. **2**, 493–507 (1995)
29. Mirkin, B.G., Fenner, T.I., Galperin, M.Y., Koonin, E.V.: Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. BMC Evol. Biol. **3, 2** (2003)
30. Nakhleh, L., Ruths, D., Wang., L.: RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer. In: Proceedings of the 11th International Computing and Combinatorics Conference, Kunming, Yunnan, pp. 84–85 (2005)
31. Page, R.D.M.: Maps between trees and cladistic analysis of historical associations among genes, organism and areas. Syst. Biol. **43**, 58–77 (1994)
32. Page, R.D.M., Charleston, M.A.: Trees within trees: phylogeny and historical associations. Trends Ecol. Evol. **13**, 356–359 (1998)
33. Robinson, D.R., Foulds, L.R.: Comparison of phylogenetic trees. Math. Biosci. **53**, 131–147 (1981)
34. Saitou, N., Nei., M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**, 406–425 (1987)
35. Sneath, P.H.A., Sokal, R.R.: Numerical Taxonomy: The Principles and Practice of Numerical Classification. W.H. Freeman, San Francisco (1973)

36. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics **22**, 2688–2690 (2006)
37. Than, C., Ruths, D., Nakhleh, L.: PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. BMC Bioinformatics **9**, 322 (2008)
38. von Haeseler, A., Churchill, G.A.: Network models for sequence evolution. J. Mol. Evol. **37**, 77–85 (1993)
39. Whidden, C., Zeh, N.: A unifying view on approximation and FPT of agreement forests. In: Proceedings of WABI'09, pp. 390–402. Springer, Berlin/Heidelberg (2009)
40. Whidden, C., Beiko, R.G., Zeh, N.: Fast FPT algorithms for computing rooted agreement forests: theory and experiments. In: Festa, P. (ed.) SEA. Lecture Notes in Computer Science, vol. 6049, pp. 14–153. Springer, Berlin (2010)
41. Zhaxybayeva, O., Lapierre, P., Gogarten, J.P.: Genome mosaicism and organismal lineages. Trends Genet. **20**, 254–260 (2004)

Part III
Measurement

Meaningful and Meaningless Statements in Landscape Ecology and Environmental Sustainability

Fred S. Roberts

Abstract The growing population and increasing pressures for development lead to challenges to life on our planet. Increasingly, we are seeing how human activities affect the natural environment, including systems that sustain life: climate, healthy air and water, arable land to grow food, etc. There is growing interest (and urgency) in understanding how changes in human activities might lead to long-term sustainability of critical environmental systems. Of particular interest are large ecological systems that affect climate, air and water, etc. *Landscape Ecology* is concerned with such systems. Understanding the challenges facing our planet requires us to summarize data, understand claims, and investigate hypotheses. To be useful, these summaries, claims, and hypotheses are often stated using metrics of various kinds, using a variety of scales of measurement. The modern theory of measurement shows us that we have to be careful using scales of measurement and that sometimes statements using such scales can be meaningless—in a very precise sense. This paper summarizes the theory of meaningful and meaningless statements in measurement and applies it to statements in landscape ecology and environmental sustainability.

Keywords Measurement • Meaningfulness • Landscape ecology • Environmental sustainability • Biodiversity • Indices

1 Introduction

The growing population and increasing pressures for development lead to challenges to life on our planet. Increasingly, we are seeing how human activities affect the natural environment, including systems that sustain life: climate, healthy air and

F.S. Roberts (✉)
DIMACS Center, Rutgers University, Piscataway, NJ 08854, USA
e-mail: froberts@dimacs.rutgers.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_18,

water, arable land to grow food, etc. There is growing interest (and urgency) in understanding how changes in human activities might lead to long-term sustainability of critical environmental systems. Of particular interest are large ecological systems that affect climate, air and water, etc. *Landscape Ecology* is concerned with such systems. Understanding the challenges facing our planet requires us to summarize data, understand claims, and investigate hypotheses. To be useful, these summaries, claims, and hypotheses are often stated using metrics of various kinds, using a variety of scales of measurement. The modern theory of measurement shows us that we have to be careful using scales of measurement and that sometimes statements using such scales can be meaningless—in a very precise sense. We will summarize the theory of meaningful and meaningless statements in measurement and apply it to statements in landscape ecology and environmental sustainability.

The modern theory of measurement was developed in part to deal with measurement in the social and biological sciences, where scales are not as readily defined as in the physical sciences. Extensive work has been done to understand scales measuring utility, noise, intelligence, etc. The theory of measurement was developed as an interdisciplinary subject, aiming at putting the foundations of measurement on a firm mathematical foundation. The theory traces its roots to work of Helmholtz in the late nineteenth century and was widely formalized in the twentieth century in such books as Krantz et al. [8], Luce et al. [10], Pfanzagl [16], Roberts [19], and Suppes et al. [33]. Measurement theory is now beginning to be applied in a wide variety of new areas. Traditional concepts of measurement theory are not well known in the landscape ecology world or in new investigations in environmental sustainability. They are finding new applications there and, in turn, problems of landscape ecology and environmental sustainability are providing new challenges for measurement theory.

We will seek to answer questions such as the following:

- Is it meaningful to say that the biodiversity of an ecosystem has increased by 10 %?
- Is the average health of forests in South Africa higher than the average health of forests in Kenya?
- For measuring the health of grasslands using vegetation indices such as leaf area index or normalized difference vegetation index, which optical instrument is best?

All of these questions have something to do with measurement. In the next section, we provide a brief introduction to the theory of measurement. Then, in Sect. 3, we formalize the concept of meaningful statement. The rest of the paper describes a variety of meaningful and meaningless statements, starting with measures of biodiversity, scales of average forest health, and vegetation index.

2 Scales of Measurement

Measurement has something to do with numbers. In the theory of measurement, we think of starting with a set A of objects that we want to measure. We shall think of a *scale of measurement* as a function f that assigns a real number $f(a)$ to each element a of A. More generally, we can think of $f(a)$ as belonging to another set B. The "representational theory of measurement" gives conditions under which a function is an *acceptable scale* of measurement. For an exposition of this theory, see, for example, Krantz et al. [8] or Roberts [19]. Following ideas of Stevens [26–28], we speak of an *admissible transformation* as a function that sends one acceptable scale into another, for example Centigrade into Fahrenheit and kilograms into pounds. In most cases, we can think of an admissible transformation as defined on the range of the scale of measurement. Suppose that f is an acceptable scale on A, taking values in B. Then a function ϕ that takes $f(a)$ into $(\phi \circ f)(a)$ is called an admissible transformation if $(\phi \circ f)(a)$ is again an acceptable scale. For example, $\phi(x) = (9/5)x + 32$ is the transformation that takes Centigrade into Fahrenheit and $\phi(x) = 2.2x$ is the transformation that takes kilograms into pounds. Stevens classified scales into types according to the associated class of admissible transformations. For instance, the class of admissible transformations of the form $\phi(x) = \alpha x$, $\alpha > 0$, defines the class of scales known as *ratio scales*. Thus, a scale f is a ratio scale if and only if every transformation $\phi(x) = \alpha x$, $\alpha > 0$, is admissible and every admissible transformation is of the form $\phi(x) = \alpha x$, $\alpha > 0$. Such transformations change the unit. Mass is an example of a ratio scale, where admissible transformations take kilograms into pounds, ounces into milligrams, grams into kilograms, etc. Time intervals are another example: we can change from years to days, from days to minutes, etc. Length is another example, with changes from meters to yards, inches to kilometers, meters to millimeters, etc. Volume is another example and so is temperature on the Kelvin scale, where there is an "absolute zero."

A second important type of scale is an *interval scale*, where the class of admissible transformations is the class of transformations of the form $\phi(x) = \alpha x + \beta$, $\alpha > 0$. Here, we can change not only the unit but also the zero point. Temperature as in Centigrade to Fahrenheit is an example of an interval scale. So is time on the calendar, where we set a zero point and can change it. For example, this is the year 2014, starting from a given year as 0.

We say a scale is an *ordinal scale* if the admissible transformations are the (strictly) monotone increasing transformations. Grades of leather, wool, etc. define ordinal scales. The Mohs scale of hardness is another ordinal scale. On this scale, every mineral gets a number between 1 and 10, but the only significance of these numbers is that a mineral with a higher number "scratches" a mineral with a lower number, and so we can use any 10 numbers rather than $1, 2, \ldots, 10$ as long as we keep the principle that a mineral assigned a higher number "scratches" one assigned a lower number. Some people feel that "preference" judgments, which lead to numbers called "utilities" in economics, only define an ordinal scale, while

some think utilities define an interval scale under certain circumstances. Subjective judgments of quality of vegetation probably also only define an ordinal scale.

We say that we have an *absolute scale* if the only admissible transformation is the identity. Counting defines an absolute scale. For definitions of some other scale types, see Roberts [19].

3 Meaningful Statements

In measurement theory, we speak of a statement as being *meaningful* if its truth or falsity is not an artifact of the particular scale values used. The following definition is due to Suppes [30] and Suppes and Zinnes [32]:

Definition. A statement involving numerical scales is meaningful if its truth or falsity is unchanged after any (or all) of the scales is transformed (independently?) by an admissible transformation.

A slightly more informal definition is the following:

Alternate Definition. A statement involving numerical scales is meaningful if its truth or falsity is unchanged after any (or all) of the scales is (independently?) replaced by another acceptable scale.

In some practical examples, for instance those involving preference judgments under the "semiorder" model, it is possible to have two scales where one cannot go from one to the other by an admissible transformation, so one has to use this alternate definition. (See Roberts [19], Roberts [21].) There is a long literature of more sophisticated approaches to meaningfulness to avoid situations where either of the above definitions may run into trouble, but we will avoid those complications here. Our emphasis is on the notion of "invariance" of truth value. Our motivation is that scales used in practice might be somewhat arbitrary, involving choices about zero points or units or the like. We would not want conclusions or decisions to be different if the arbitrary choices made are changed in some "admissible" way.

To start, let us consider the following statement:

Statement S. "The duration of the most recent drought in a given ecological reserve was three times the duration of the previous drought."

Is this meaningful? We have a ratio scale (time intervals) and we consider the statement:.

$$f(a) = 3f(b). \tag{1}$$

This is meaningful if f is a ratio scale. For, an admissible transformation is $\phi(x) = \alpha x, \alpha > 0$. We want Eq. (1) to hold iff

$$(\phi \circ f)(a) = 3(\phi \circ f)(b). \tag{2}$$

But Eq. (2) becomes

$$\alpha f(a) = 3\alpha f(b) \tag{3}$$

and (1) iff (3) since $\alpha > 0$. Thus, the statement S is meaningful.

Consider next the statement:

Statement T. "The high temperature in a given ecological reserve in 2012 was 2 per cent higher than it was in 1912."

Is this meaningful? This is the statement

$$f(a) = 1.02 f(b).$$

This is meaningless. It could be true with Fahrenheit and false with Centigrade, or vice versa. In general, for ratio scales, it is meaningful to compare ratios:

$$f(a)/f(b) > f(c)/f(d).$$

For interval scales, it is meaningful to compare intervals:

$$f(a) - f(b) > f(c) - f(d).$$

For ordinal scales, it is meaningful to compare size:

$$f(a) > f(b).$$

Sometimes in ecology, we try to weigh samples. We might have two equal size baskets, one containing feathers and one containing (elephant) tusks. Consider the claim:

Statement W. "The total weight of my basket of feathers is 1000 times that of my basket of tusks."

Is this statement meaningful? Yes, since it involves ratio scales and is presumably false no matter what unit is used to measure weight. The point is that meaningfulness is different from truth. It has to do with what kinds of assertions it makes sense to make, which assertions are not accidents of the particular choice of scale (units, zero points) in use.

4 Biodiversity

Next we ask if it is meaningful to say that the biodiversity of an ecosystem has increased by 10 %. Evidence about the health of ecosystems is often obtained by measuring the biodiversity. Loss of biodiversity is considered an indicator of

declining health of an ecosystem and there is great concern that climate change and other environmental stressors—natural and man-made—are leading to such a loss. One way of measuring progress in controlling the unwanted environmental effects of human activities—effects of human systems on natural systems—is to determine the extent to which the loss of biodiversity has been controlled. An index of biodiversity allows us to set specific goals and measure progress toward them. The 1992 Convention on Biological Diversity (CBD) (http://www.biodiv.org) set the goal that, by 2010, we should achieve a significant reduction of the current state of biodiversity loss at the global, regional, and national level [34]. How can we tell if we have achieved this goal? We need to be able to measure biodiversity.

There have been hundreds of papers attempting to define biodiversity precisely. Traditional approaches consider two basic determinants of biodiversity: *Richness* is the number of species and *evenness* is the extent to which species are equally distributed [11]. These concepts assume that all species are equal, that all individuals are equal (we disregard differences in size, health, etc.), and that spatial distribution is irrelevant. These may not be appropriate assumptions. We shall concentrate here on the notion of evenness, which is based on ideas going back in the economic literature to the work of Gini [3, 4] on measures of even income distribution and of Dalton [1] on measures of inequality. Some measures of biodiversity or evenness go back to work in communication theory, in particular the work of Shannon [24] on entropy in information theory.

Let S be the number of species in an ecosystem and x_i be the number of individuals of species i found (the *abundance* of species i). In some cases, x_i is not a number, but some measure of biomass, e.g., grams per square meter. The vector $\mathbf{x} = (x_1, x_2, \ldots, x_S)$ is called the *abundance vector* and we seek a measure of evenness $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_S)$. We shall take $f(\mathbf{x})$ to be low if very even, high if very uneven. Finally, let a_i be the proportion of the population represented by species i, i.e., $a_i = x_i / \sum_j x_j$. In the literature, there are many proposed measures of evenness. We give a few examples. The *Simpson index* [25] is given by $\lambda = \sum_i {a_i}^2$. It measures the probability that any two individuals drawn at random from an infinite population will belong to the same species. The *Shannon–Wiener Diversity Index* is given by $-\sum_i a_i \ln(a_i)$. In information theory, the negative of this index is called the Shannon entropy. The Shannon entropy is maximized if each x_i is the same, so the Shannon–Wiener Diversity Index is minimized in this case.

Let us consider the statement that the biodiversity of an ecosystem has increased by 10 % as the following:

Statement E. "The evenness of an ecosystem has increased by 10%."

If x_i is the number of individuals of species i, then we have an absolute scale and the only admissible transformation of scale is the identity, so a_i does not change and neither does either of the indices of evenness we are looking at. So, the statement is meaningful. However, what if x_i is the biomass of species i, for example kilograms of i per square meter? Both mass and length are ratio scales, so we can change, for example, from kilograms per square meter to grams per square centimeter, and so on. What happens if we multiply mass by a constant α and length by a constant β?

Let y_i be the new abundance value and b_i be the new abundance proportion for species i. We have

$$y_i = (\alpha/\beta^2)x_i, \tag{4}$$

so

$$b_i = y_i / \sum_j y_j = (\alpha/\beta^2)x_i / \sum_j (\alpha/\beta^2)x_j = a_i. \tag{5}$$

It follows that neither the Simpson Index nor the Shannon Index changes after we change units, and so the Statement E is meaningful.

5 Averaging Judgments of Forest Health

Suppose we study two groups of forests, one in South Africa and one in Kenya. Let $f(a)$ be the health of forest a as judged by an "expert" on a subjective forest health scale using values 1–5 or 1–6, as is sometimes done. Suppose that data suggests that the average health of the forests in South Africa is higher than that of the forests in Kenya. Is this meaningful? Let $a_1, a_2, \ldots a_n$ be forests in the South African group and $b_1, b_2, \ldots, b_m$ be forests in the Kenyan group. Note that m could be different from n. Then we are (probably) asserting that

$$\frac{1}{n}\sum_{i=1}^{n} f(a_i) > \frac{1}{m}\sum_{i=1}^{m} f(b_i). \tag{6}$$

We are comparing arithmetic means. The statement (6) is meaningful if and only if under admissible transformation ϕ, (6) holds if and only if

$$\frac{1}{n}\sum_{i=1}^{n} (\phi \circ f)(a_i) > \frac{1}{m}\sum_{i=1}^{m} (\phi \circ f)(b_i) \tag{7}$$

holds. If forest health defines a ratio scale, then (7) is the same as

$$\frac{1}{n}\sum_{i=1}^{n} \alpha f(a_i) > \frac{1}{m}\sum_{i=1}^{m} \alpha f(b_i), \tag{8}$$

for some positive α. Certainly (6) holds if and only if (8) does, so (6) is meaningful. This kind of comparison would work if we were simply comparing biomass of forests.

Note that (6) is still meaningful if f is an interval scale. For instance, we could be comparing utility or worth of a forest (e.g., in terms of "ecosystem services")

$f(a)$. Some economists think that in some cases, utility defines an interval scale. It is meaningful to assert that the average health of the first group is higher than the average health of the second group. To see why, note that (6) is equivalent to

$$\frac{1}{n}\sum_{i=1}^{n}[\alpha f(a_i)+\beta] > \frac{1}{m}\sum_{i=1}^{m}[\alpha f(b_i)+\beta],$$

where $\alpha > 0$.

However, (6) is easily seen to be meaningless if f is just an ordinal scale. To show that comparison of arithmetic means can be meaningless for ordinal scales, note that we are asking experts for a subjective judgment of forest health. Suppose that $f(a)$ is measured on a 5-point scale: 5 = very healthy, 4 = healthy, 3 = neutral, 2 = unhealthy, 1 = very unhealthy. In such a scale, the numbers may not mean anything; only their order matters. Suppose that group 1 has three members with scores of 5, 3, and 1, for an average of 3, while group 2 has three members with scores of 4, 4, and 2 for an average of 3.33. Then the average score in group 2 is higher than the average score in group 1. On the other hand, suppose we consider the admissible transformation ϕ defined by $\phi(5) = 100$, $\phi(4) = 75$, $\phi(3) = 65$, $\phi(2) = 40$, $\phi(1) = 30$. Then after transformation, members of group 1 have scores of 100, 65, 30, with an average of 65, while those in group 2 have scores of 75, 75, 40, with an average of 63.33. Now, group 1 has a higher average score. Which group had a higher average score? The answer clearly depends on which version of the scale is used. Of course, one can argue against this kind of example. As Suppes [31] remarks in the case of a similar example having to do with grading apples in four ordered categories, "surely there is something quite unnatural about this transformation" ϕ. He suggests that "there is a strong natural tendency to treat the ordered categories as being equally spaced." However, if we require this, then the scale is not an ordinal scale according to our definition. Not every strictly monotone increasing transformation is admissible. Moreover, there is no reason, given the nature of the categories, to feel that this is demanded in our example. In any case, the argument is not with the precept that we have stated, but with the question of whether the five-point scale we have given is indeed an ordinal scale as we have defined it. To complete this example, let us simply remark that comparison of medians rather than arithmetic means is meaningful with ordinal scales: The statement that one group has a higher median than another group is preserved under admissible transformation.

Let us return to forest health, but now suppose that each of n observers is asked to rate each of a collection of forests as to their relative health. Alternatively, suppose we rate forests on different criteria or against different benchmarks. (A similar analysis applies with performance ratings, importance ratings, etc.) Let $f_i(a)$ be the rating of forest a by expert i (or under criterion i). Is it meaningful to assert that the average rating of forest a is higher than the average rating of forest b? A similar question arises in expert-judged ratings of health of individual species, quality of water in a stream, severity of pollution, etc.

We are now considering the statement

$$\frac{1}{n}\sum_{i=1}^{n} f_i(a) > \frac{1}{n}\sum_{i=1}^{n} f_i(b). \tag{9}$$

Note in contrast to statement (6) that we have the same number of terms in each sum and that the subscript is now on the scale value f rather than on the alternative a or b. If each f_i is a ratio scale, we then ask whether or not (9) is equivalent to

$$\frac{1}{n}\sum_{i=1}^{n} \alpha f_i(a) > \frac{1}{n}\sum_{i=1}^{n} \alpha f_i(b),$$

$\alpha > 0$. This is clearly the case.

However, we have perhaps gone too quickly. What if $f_1, f_2, \ldots, f_n$ have independent units? In this case, we want to allow independent admissible transformations of the f_i. Thus, we must consider

$$\frac{1}{n}\sum_{i=1}^{n} \alpha_i f_i(a) > \frac{1}{n}\sum_{i=1}^{n} \alpha_i f_i(b), \tag{10}$$

all $\alpha_i > 0$. It is easy to find $\alpha_i's$ for which (9) holds but (10) fails. Thus, (9) is meaningless. Does it make sense to consider different α_i? It certainly does in some contexts. Consider the case where the alternatives are animals in an ecosystem and one expert measures their health in terms of their weight gain while a second measures it in terms of their height gain.

The conclusion is that we need to be careful when comparing arithmetic mean ratings, even when we are using ratio scales. Norman Dalkey (personal communication) was the first person to point out to the author that, in many cases, it is safer to use geometric means, a conclusion which by now is "folklore." For, consider the comparison

$$\sqrt[n]{\prod_{i=1}^{n} f_i(a)} > \sqrt[n]{\prod_{i=1}^{n} f_i(b)}. \tag{11}$$

If all $\alpha_i > 0$, then (11) holds if and only if

$$\sqrt[n]{\prod_{i=1}^{n} \alpha_i f_i(a)} > \sqrt[n]{\prod_{i=1}^{n} \alpha_i f_i(b)}.$$

Thus, if each f_i is a ratio scale, then even if experts change the units of their rating scales independently, the comparison of geometric means is meaningful even though

the comparison of arithmetic means is not. An example of an application of this observation is the use of the geometric mean by Roberts [17,18]. The problem arose in a study of air pollution and energy use in commuter transportation. A preliminary step in the model building involved the choice of the most important variables to consider in the model. Each member of a panel of experts estimated the relative importance of variables using a procedure called magnitude estimation. (Here, the most important variable is given a score of 100, a variable judged half as important is given a score of 50, and so on.) There is a strong body of opinion that magnitude estimation leads to a ratio scale, much of it going back to Stevens [29]. (See the discussion in Roberts [19, pp. 179–180].) How then should we choose the most important variables? By the discussion above, it is "safer" to combine the experts' importance ratings by using geometric means and then to choose the most important variables as those having the highest geometric mean relative importance ratings, than it is to do this by using arithmetic means. That is why Roberts [17, 18] used geometric means.

6 Evaluation of Alternative Optical Instruments for Measuring Vegetation Indices

Various indices have been developed to characterize type, amount, and condition of vegetation present. Remote sensing is often used for this purpose. Among the indices of interest are the leaf area index and the normalized difference vegetation index, both based on spectral reflectance [7]. Recent developments have provided a variety of new types of optical remote sensing equipment for estimating reflectance characteristics and thus calculating indices (see e.g., [36]). What if we want to compare alternative remote sensing devices that are candidates for this use? How might we do it?

One common procedure for comparing alternative instruments, machines, treatments, etc. is the following. A number of instruments are compared on different criteria/benchmarks. Their scores on each criterion are normalized relative to the score of one of the instruments. The normalized scores of an instrument are combined by some averaging procedure and average scores are compared. If the averaging is the arithmetic mean, then consider the statement:

Statement N. "One instrument has a higher arithmetic mean normalized score than another instrument."

Statement N is meaningless: The instrument to which scores are normalized can determine which has the higher arithmetic mean. Similar methods are used in comparing performance of alternative computer systems or other types of machinery. To illustrate, consider a number of potential criteria for optical instruments for measuring vegetation indices: Accuracy on cloudy days, accuracy with low-stature vegetation, accuracy for extremely diverse forests, ease of use, reliability, etc.

Table 1 Score of instrument i on criterion j

Instrument/Criterion	A	B	C	D	E
I	417	83	66	39,449	772
II	244	70	153	33,527	368
III	134	70	135	66,000	369

Table 2 Normalizing relative to instrument I

Instrument/Criterion	A	B	C	D	E	Arithmetic mean	Geometric mean
I	1.00	1.00	1.00	1.00	1.00	1.00	1.00
II	.59	.84	2.32	.85	.48	1.01	.86
III	.32	.85	2.05	1.67	.45	1.07	.84

Table 3 Normalizing relative to instrument II

Instrument/Criterion	A	B	C	D	E	Arithmetic mean	Geometric mean
I	1.71	1.19	.43	1.18	2.10	1.32	1.17
II	1.00	1.00	1.00	1.00	1.00	1.00	1.00
III	.55	1.00	1.88	1.97	1.08	1.07	.99

Table 1 shows three instruments I, II, III and five criteria A, B, C, D, E, with the i, j entry giving the score of the ith treatment on the jth criterion. Table 2 shows the score of each instrument normalized relative to treatment I, i.e., by dividing by instrument I's score. Thus, for example, the 1,2 entry is $83/83 = 1$, while the 2,2 entry is $70/83 = .84$. The arithmetic means of the normalized scores in each row are also shown in Table 2. We conclude that instrument III is best.

However, let us now normalize relative to Instrument II, obtaining the normalized scores of Table 3. Based on the arithmetic mean normalized scores of each row shown in Table 3, we now conclude that Instrument I is best. So, the conclusion that a given instrument is best by taking arithmetic mean of normalized scores is meaningless in this case: Statement N is meaningless.

The numbers in this example are taken from Fleming and Wallace [2], with data from Heath [6], and represent actual scores of alternative "instruments" in a computing machine application.

Sometimes, geometric mean is helpful. The geometric mean normalized scores of each row are shown in Tables 2 and 3. Note that in each case, we conclude that Instrument I is best. In this situation, it is easy to show that the conclusion that a given instrument has highest geometric mean normalized score is a meaningful conclusion. It is even meaningful to assert something like: A given instrument has geometric mean normalized score 20 % higher than another instrument.

Fleming and Wallace give general conditions under which comparing geometric means of normalized scores is meaningful. We have now given several examples where comparing geometric means leads to meaningful conclusions while comparing arithmetic means does not. However, there are situations where comparing

arithmetic means leads to meaningful conclusions and comparing geometric means does not. It is a research area in measurement theory, with a long history and large literature, to determine what averaging procedures make sense in what situations. For some further details on this topic, and in particular for an example where arithmetic mean comparison is meaningful while geometric mean is not, see Roberts [23].

The message from measurement theory is: Do not perform arithmetic operations on data without paying attention to whether the conclusions you get are meaningful.

7 Optimization Problems in Landscape Ecology

Raster datasets represent geographic features by dividing the world into discrete square or rectangular cells laid out in a grid. Each cell has a characteristic value that is used to represent some characteristic of that location. As noted by Zettenberg [37], a GIS raster can be seen as a network, with grid cells as nodes and a link (edge) from each cell to its vertical, horizontal, and diagonal neighbors. The links might have weights or costs on them. According to Zettenberg, the least-cost path in between two nodes can represent a "geodesic path" between two points "(approximated by grid cells) on a projected surface. ... Even though the straight line Euclidean distance is a lot shorter, it may be *functionally* shorter for example to follow a detour along a preferred habitat." As Zettenberg also says, within the raster, the "cost-distance value at any point (i.e., grid cell) is the least-cost distance from that point to the closest specified source point." Sometimes we seek "patches" made up of cells with cost-distance value below some threshold that corresponds to some ecologically relevant value. The problem of finding the shortest distance between two nodes in a network (where the "length" of a path is the sum of weights on edges in it) is a widely studied problem in operations research and there are very efficient algorithms for solving it. The shortest path problem occurs widely in practice. In the USA, just one agency of the US Department of Transportation in the federal government has applied algorithms to solve this problem literally billions of times a year [5].

Consider a simple network with nodes x, y, and z and edges from x to y with weight 2, y to z with weight 4, and x to z with weight 15. What is the shortest path from x to z in this network? The shortest path is the path that goes from x to y to z, with a total "length" of 6. The alternative path that goes directly from x to z has total "length" 15. Is the conclusion that x to y to z is the shortest path a meaningful conclusion?

The conclusion is meaningful if the weights on edges define a ratio scale, as they do if they are physical distances or monetary amounts. However, what if they define an interval scale? This could happen if the weights are utilities or values, rather than dollar amounts or physical lengths. As noted earlier, utilities might be defined on interval scales. If the weights define an interval scale, consider the admissible transformation $\phi(x) = 3x + 100$. Now the weights change to 106 on the edge from

x to y, 112 on the edge from y to z, and 145 on the edge from x to z. We conclude that going directly from x to z is the shortest path. The original conclusion was meaningless.

The shortest path problem can be formulated as a linear programming problem. Thus, the conclusion that A is the solution to a linear programming problem can be meaningless if cost parameters are measured on an interval scale. Note that linear programming is widely used in landscape ecology as well as in other areas of application. For example, it is used to determine optimal inventories of equipment, assignments of researchers to projects, optimization of the size of an ecological reserve, amount to invest in preventive treatments, etc.

Another very important practical combinatorial optimization problem is the minimum spanning tree problem. Given a connected, weighted graph or network, we ask for the spanning tree with total sum of costs or weights as small as possible. (A *spanning tree* is a tree that includes all the nodes of the network.) This problem has applications in the planning of large-scale transportation, communication, and distribution networks, among other things. Minimum spanning trees arise in landscape ecology in the following way. Following Urban and Keitt [35], consider a landscape of habitat patches. Build a graph whose nodes are the patches, with an edge between patches if there is some "ecological flux" between them, e.g., via dispersal or material flow. Put weights on the edges to reflect flow rates or dispersal probabilities. Next, the patches are rated in terms of their "importance." We consider patterns of habitat loss and degradation. In a simplified model, we remove patches in entirety one at a time, i.e., remove available habitat gradually, one patch at a time. This amounts to removing one node from the graph at a time. We study preservation of species by asking how much habitat must be removed before that species becomes extinct (at least in the system being modeled).

Urban and Keitt studied the following patch-removal algorithm: Find a minimum spanning tree that has a "leaf" (node with only one neighbor) of smallest importance and remove the patch corresponding to that leaf. Then repeat the process on the remaining graph. Urban and Keitt studied this process for the Mexican Spotted Owl. In 1993, this subspecies was listed as threatened under the Endangered Species Act in the USA. Habitat distribution for this species is highly fragmented in the US Southwest. By using this patch-removal algorithm, Urban and Keitt found in simulation models that the Mexican Spotted Owl population actually increased until nearly all the habitat was removed. By way of contrast, if patches were removed in random order, the owl population declined dramatically as habitat was removed. Urban and Keitt explain their algorithm by noting that the spanning tree "maintains the integrity of the landscape by not only providing large core populations, but also by providing dispersal routes between core habitats."

It is natural to ask if the conclusion that a given set of edges defines a minimum spanning tree is meaningful. (In Urban and Keitt's work, determining the scale type of the edge-weights is a rather complex issue.) It is surprising to observe that even if the weights on the edges define only an ordinal scale, then the conclusion is meaningful. This is not a priori obvious. However, it follows from the fact that the well-known algorithm known as Kruskal's algorithm or the greedy algorithm

gives a solution. In Kruskal's algorithm [9, 13], we order edges in increasing order of weight and then examine edges in this order, including an edge if it does not form a cycle with edges previously included. We stop when all nodes are included. Since any admissible transformation will not change the order in which edges are examined in this algorithm, the same solution will be produced.

Many practical decision-making problems in landscape ecology, environmental sustainability, and other fields involve the search for an optimal solution as in the shortest path and minimum spanning tree problems. Little attention is paid to the possibility that the conclusion that a particular solution is optimal may be an accident of the way that things are measured. For the beginnings of the theory of meaningfulness of conclusions in combinatorial optimization, see Mahadev et al. [12], Pekeč [14, 15], and Roberts [20–22].

There is much more analysis of a similar nature in the field of landscape ecology or the study of sustainable environments that can be done with the principles of measurement theory. The issues involved present challenges both for theory and for application.

Acknowledgments The author gratefully acknowledges the support of the National Science Foundation under grant number DMS-0829652 to Rutgers University. A number of ideas and some of the examples and language in this paper are borrowed from my papers Roberts [21, 23], which explore meaningful and meaningless statements in operations research and in epidemiology and public health, respectively. The author gratefully and thankfully acknowledges the many stimulating and fruitful scientific interchanges with Boris Mirkin over a period of many years, and wishes him many years of continued good health and success.

References

1. Dalton, H.: The measurement of inequality of incomes. Econ. J. **30**, 348–361 (1920)
2. Fleming, P.J., Wallace, J.J.: How not to lie with statistics: the correct way to summarize benchmark results. Commun. ACM **29**, 218–221 (1986)
3. Gini, C.: Il diverso accrescimento delle classi sociali e la concentrazione della richezza. Giornale degli Economist, serie II, **2** (1909)
4. Gini, C.: Variabilite mutabilita. Studi Economicoaguridic della Facotta di Giurisprudenza dell Univ. di Cagliari III. Parte II. (1912)
5. Goldman, A.J.: Discrete mathematics in government. Lecture presented at SIAM Symposium on Applications of Discrete Mathematics, Troy, NY, June 1981
6. Heath, J.L.: Re-evaluation of RISC I. Comput. Archit. News **12**, 3–10 (1984)
7. Jackson, R.D., Huete, A.R.: Interpreting vegetation indices. Prev. Vet. Med. **11**, 185–200 (1991)
8. Krantz, D.H., Luce, R.D., Suppes, P., Tversky, A.: Foundations of Measurement, vol. I. Academic, New York (1971)
9. Kruskal, J.B.: On the shortest spanning tree of a graph and the traveling salesman problem. Proc. Am. Math. Soc. **7**, 48–50 (1956)
10. Luce, R.D., Krantz, D.H., Suppes, P., Tversky, A.: Foundations of Measurement, vol. III. Academic, New York (1990)

11. Magurran, A.E.: Ecological Diversity and its Measurement. Chapman & Hall, London (1991)
12. Mahadev, N.V.R., Pekeč, A., Roberts, F.S.: On the meaningfulness of optimal solutions to scheduling problems: can an optimal solution be non-optimal? Oper. Res. **46**(Suppl.), S120–S134 (1998)
13. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs (1982)
14. Pekeč, A.: Limitations on conclusions from combinatorial optimization. Ph.D. thesis, Department of Mathematics, Rutgers University (1996)
15. Pekeč, A.: Scalings in linear programming: necessary and sufficient conditions for invariance. Center for Basic Research in Computer Science (BRICS), Technical report RS-96-50 (1996)
16. Pfanzagl, J.: Theory of Measurement. Wiley, New York (1968)
17. Roberts, F.S.: Building an energy demand signed digraph I: choosing the nodes. Rept. $927/1-NSF$. April. The RAND Corporation, Santa Monica (1972)
18. Roberts, F.S.: Building and analyzing an energy demand signed digraph. Environ. Plan. **5**, 199–221 (1973)
19. Roberts, F.S.: Measurement Theory, with Applications to Decisionmaking, Utility, and the Social Sciences. Addison-Wesley, Reading (1979). Digital Reprinting (2009). Cambridge University Press, Cambridge
20. Roberts, F.S.: Meaningfulness of conclusions from combinatorial optimization. Discrete Appl. Math. **29**, 221–241 (1990)
21. Roberts, F.S.: Limitations on conclusions using scales of measurement. In: Pollock, S.M., Rothkopf, M.H., Barnett, A. (eds.) Operations Research and the Public Sector. Handbooks in Operations Research and Management Science, vol. 6, pp. 621–671. North-Holland, Amsterdam (1994)
22. Roberts, F.S.: Meaningless statements. In: Contemporary Trends in Discrete Mathematics. DIMACS Series, vol. 49, pp. 257–274. American Mathematical Society, Providence (1999)
23. Roberts, F.S.: Meaningful and meaningless statements in epidemiology and public health. In: Berglund, B., Rossi, G.B., Townsend, J., Pendrills, L. (eds.) Measurements with Persons, pp. 75–95. Taylor and Francis, New York (2012)
24. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
25. Simpson, E.H.: Measurement of diversity. Nature **163**, 688 (1949)
26. Stevens, S.S.: On the theory of scales of measurement. Science **103**, 677–680 (1946)
27. Stevens, S.S.: Mathematics, measurement, and psychophysics. In: Stevens, S.S. (ed.) Handbook of Experimental Psychology, pp. 1–49. Wiley, New York (1951)
28. Stevens, S.S.: Measurement, psychophysics, and utility. In: Churchman, C.W., Ratoosh, P. (eds.) Measurement: Definitions and Theories, pp. 18–63. Wiley, New York (1959)
29. Stevens, S.S.: Ratio scales of opinion. In: Whitla, D.K. (ed.) Handbook of Measurement and Assessment in Behavioral Sciences. Addison-Wesley, Reading (1968)
30. Suppes, P.: Measurement, empirical meaningfulness and three-valued logic. In: Churchman, C.W., Ratoosh, P. (eds.) Measurement: Definitions and Theories, pp. 129–143. Wiley, New York (1959)
31. Suppes, P.: Replies. In: Bogdan, R.J. (ed.) Patrick Suppes, pp. 207–232. Reidel, Dordrecht (1979)
32. Suppes, P., Zinnes, J.: Basic measurement theory. In: Luce, R.D., Bush, R.R., Galanter, E. (eds.) Handbook of Mathematical Psychology, vol. 1, pp. 1–76. Wiley, New York (1963)
33. Suppes, P., Krantz, D.H., Luce, R.D., Tversky, A.: Foundations of Measurement, vol. II. Academic, New York (1989)
34. UNEP: Report of the Sixth Meeting of the Conference of the Parties to the Convention on Biological Diversity (UNEP/CBD/COP/6/20) (2002)

35. Urban, D., Keitt, T.: Landscape connectivity: a graph-theoretic perspective. Ecology **82**, 1205–1218 (2001)
36. van Wijk, M.T., Williams, M.: Optical instruments for measuring leaf area index in low vegetation: application in Arctic ecosystems. Ecol. Appl. **15**, 1462–1470 (2005)
37. Zettenberg, A.: Network based tools and indicators for landscape ecological assessments, planning, and design. Licentiate Thesis, KTH-Environmental Management and Assessment Research Group, Department of Land and Water Resources Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden (2009)

Nearest Neighbour in Least Squares Data Imputation Algorithms for Marketing Data

Ito Wasito

Abstract Marketing research operates with multivariate data for solving such problems as market segmentation, estimating purchasing power of a market sector, modeling attrition. In many cases, the data collected or supplied for these purposes may have a number of missing entries.

The paper is devoted to an empirical evaluation of method for imputation of missing data in the so-called nearest neighbour of least-squares approximation approach, a non-parametric computationally efficient multidimensional technique. We make contributions to each of the two components of the experiment setting: (a) An empirical evaluation of the nearest neighbour in least-squares data imputation algorithm for marketing research (b) experimental comparisons with expectation–maximization (EM) algorithm and multiple imputation (MI) using real marketing data sets. Specifically, we review "global" methods for least-squares data imputation and propose extensions to them based on the nearest neighbours (NN) approach. It appears that NN in the least-squares data imputation algorithm almost always outperforms EM algorithm and is comparable to the multiple imputation approach.

Keywords Least squares • Nearest neighbours • Singular value decomposition • Missing data • Marketing data

1 Introduction

Marketing research operates with multivariate data for solving such problems as market segmentation, estimating purchasing power of a market sector and modeling attrition [7]. In many cases, the data collected or supplied for this may have a number

I. Wasito (✉)
Faculty of Computer Science, University of Indonesia, Kampus UI, Depok 16424, Indonesia
e-mail: ito.wasito@cs.ui.ac.id

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_19,

of missing entries. For example, in the popular commercial demographic databases like Infobase the level of population of some fields (variables) falls down to 40 %. The probability for each individual data point to be missed may be considered random: our experience did not show any significant correlation between missing entries in different fields [23].

Although some statistical methods, especially those in regression analysis, provide for built-in treatment of missing entries, the problems in marketing research cannot be reduced to application of individual methods and involve complex processing [11]. The filling of missing values or even correction of mistakes in data is not a self-dependent goal. This is just prelude to further going statistical analysis. And if this statistical analysis is used to fill out the data in order to use corrected data again for almost the same purposes, there is always a danger, that it will bring some element of tautology in the whole process.

Consider a simple example. What is happening in widely used method, when missing values replace by average value, counted by existing ones, well-known effect is, that new average after that replacement is equal to old one. It is obvious that for variable with replaced value it did not give new information: if analysis of data is limited just by analysis of means, it is just trivial (the same value remains); if it goes further to the level of analysis of deviation, correlation and so on, just losses are obvious, because this replacement may distort all those parameters in any (unknown) proportion. This proportion, in principle, may be more definite just under some kind of assumptions about mechanism of data generation, distribution, etc., but even in that case just general estimation of distortion, not caused by each particular replacement.

This example shows one important aspect of missing values problem, which is usually ignored: it is important to fill them out (versus case wise deletion). There could be two different purposes for that [11, 12, 18].

(1) The filling of missing value is considered as procedure of "approximation", where new found value is important itself for this particular object. It makes sense in situations like regression estimation on each object. (2) The filling of missing data is good, because it allows to get back that part of information about non-missing values, which otherwise will be case-wise deleted. This second aspect is really positive in many situations, but it was not investigated at all.

The data we used is a sample from the typical database of the large manufacturer and devoted to the problem of retention of existing customers. There are many variables describing customers behavior and service features, and there is a target binary variable ("refused the service or not"). The problem is to create a satisfactory recognition rule(-s) to predict those who will cancel the service agreement. The data set and the problem formulated are quite typical for many applications, and in that sense the reconstruction of missing values for such a data is of big practical interest.

In [22, 23] the authors reviewed and compared various least-squares-based algorithms and proposed a number of their modifications involving the nearest neighbourhood methods, then carried out a series of computational experiments involving uniformly random missing entries. In our experiments we separately generate a complete data matrix and a set of entries that are considered missing in it.

This design enables us, for any data set and pattern of missing data, to compare the imputed values with those originally generated: the smaller the difference, the better the method. According to experiments showed in [22], the two different data models lead to different results. With the unidimensional data generator, the best imputation methods are those using just one factor. Nearest neighbour-based modifications do not improve results in this case, even at high levels of noise. In contrast, with data generated according to the Gaussian mixture distribution, methods involving the nearest neighbours are the best.

In this paper we extend the study published in [22,23] for marketing data research as explained above.

The paper is organized as follows. Section 2 gives a brief description of the global least-squares imputation methods and also our NN versions of the imputation methods will be described. In Sect. 3, we will review EM-based algorithm for data imputation. Section 4 provides for the setting of experiments and results discussions. and Sect. 5 concludes the paper.

2 Nearest Neighbour in the Least-Squares Data Imputation Algorithm

Before describing the algorithm, first, some theoretical background of iterative SVD and the iterative majorization least-squares (IMLS) algorithm will be introduced.

2.1 Notation

The data is considered in the format of a matrix $\mathbf{X}$ with N rows and n columns. The rows are assumed to correspond to entities (observations) and columns to variables (features). The elements of $\mathbf{X}$ are denoted by x_{ik} $(i = 1, \ldots, N, k = 1, \ldots, n)$. The situation in which some entries (i, k) in $\mathbf{X}$ are missed is modeled with an additional matrix $\mathbf{M} = (m_{ik})$ where $m_{ik} = 0$ if the (i, k)-th entry is missed and $m_{ik} = 1$, otherwise.

The matrices and vectors are denoted with boldface letters. A vector is always considered as a column; thus, the row vectors are denoted as transposes of the column vectors. Sometimes we show the operation of matrix multiplication with symbol $*$.

2.2 Iterative Singular Value Decomposition

Let us describe the concept of singular value decomposition of a matrix (SVD) in terms of a bilinear model for factor analysis of data. This model assumes the

existence of a number $p \geq 1$ of hidden factors that underlie the observed data as follows:

$$x_{ik} = \sum_{t=1}^{p} c_{tk} z_{it} + e_{ik}, \quad i = 1, \ldots N, \quad k = 1, \ldots, n. \tag{1}$$

The vectors $\mathbf{z}_t = (z_{it})$ and $\mathbf{c}_t = (c_{tk})$ are referred to as factor t scores for entities $i = 1, \ldots, N$ and factor loadings for variables $k = 1, \ldots, n$, respectively ($t = 1, \ldots, p$) [5,9,13]. Values e_{ik} are residuals that are not explained by the model and should be made as small as possible.

To find approximating vectors $\mathbf{c}_t = (c_{tk})$ and $\mathbf{z}_t = (z_{it})$, one minimizes the least-squares criterion:

$$L_2 = \sum_{i=1}^{N} \sum_{k=1}^{n} \left(x_{ik} - \sum_{t=1}^{p} c_{tk} z_{it} \right)^2 \tag{2}$$

It is proven that minimizing criterion (2) can be done with the following one-by-one strategy, which is, basically, the contents of the method of principal component analysis, one of the major data mining techniques [5, 9] as well as the so-called power method for SVD.

According to this strategy, computations are carried out iteratively. At each iteration t, $t = 1, \ldots, p$, only one factor is sought. The criterion to be minimized at iteration t is

$$l_2(\mathbf{c}, \mathbf{z}) = \sum_{i=1}^{N} \sum_{k=1}^{n} (x_{ik} - c_k z_i)^2 \tag{3}$$

with respect to condition $\sum_{k=1}^{n} c_k^2 = 1$. It is well known that it is the singular triple $(\mu, \mathbf{z}, \mathbf{c})$ such that $\mathbf{Xc} = \mu\mathbf{z}$ and $\mathbf{X^T z} = \mu\mathbf{c}$ with $\mu = \sqrt{\sum_{i=1}^{N} z_i^2}$, the maximum singular value of $\mathbf{X}$, which solves the problem. The found vectors $\mathbf{c}$ and $\mathbf{z}$ are stored as $\mathbf{c}_t$ and $\mathbf{z}_t$ and next iteration $t + 1$ is performed. The matrix $\mathbf{X} = (x_{ik})$ changes from iteration t to iteration $t + 1$ by subtracting the found solution according to the rule $x_{ik} \leftarrow x_{ik} - c_{tk} z_{ti}$.

To minimize (3), the method of alternating minimization can be utilized. This method also works iteratively. Each iteration proceeds in two steps: (1) given a vector (c_k), find optimal (z_i); (2) given (z_i), find optimal (c_k), which can be done according to equations:

$$z_i = \frac{\sum_{k=1}^{n} x_{ik} c_k}{\sum_{k=1}^{n} c_k^2} \tag{4}$$

and

$$c_k = \frac{\sum_{i=1}^{N} x_{ik} z_i}{\sum_{i=1}^{N} z_i^2} \tag{5}$$

that follow from the first-order optimality conditions.

This can be wrapped up by the following algorithm for finding a pre-specified number p of singular values and vectors.

Iterative SVD Algorithm

0. Set number of factors p *and specify* $\epsilon > 0$, *a precision threshold.*
1. Set iteration number t=1.
2. Initialize $\mathbf{c}^*$ *arbitrarily and normalize it. (Typically, we take* $\mathbf{c}^{*\prime} = (1 \ldots, 1)$.*)*
3. Given $\mathbf{c}^*$, *calculate* $\mathbf{z}$ *according to (4).*
4. Given $\mathbf{z}$ *from step 3, calculate* $\mathbf{c}$ *according to (5) and normalize it.*
5. If $||\mathbf{c} - \mathbf{c}^*|| < \epsilon$, *go to 6;*
otherwise put $\mathbf{c}^* = \mathbf{c}$ *and go to 3.*
6. Set $\mu = ||\mathbf{z}||$, $\mathbf{z_t} = \mathbf{z}$, *and* $\mathbf{c_t} = \mathbf{c}$.
7. If $t == p$, *end; otherwise, update* $x_{ik} = x_{ik} - c_{tk} z_{tk}$, *set* $t = t + 1$ *and go to step 2.*

Note that $\mathbf{z_t}$ is not normalized in the described version of the algorithm, which implies that its norm converges to the singular value μ_t indeed. This method always converges if the initial $\mathbf{c}$ does not belong to the subspace already taken into account in the previous singular vectors.

2.3 IMLS Algorithm

This method is an example of application of the general idea that the weighted least-squares minimization problem can be addressed as a series of non-weighted least-squares minimization problems with iteratively adjusting found solutions according to a so-called majorization function [4, 6]. In this framework, Kiers [10] developed the following algorithm that in its final form can be formulated without any concept beyond those previously specified. The algorithm starts with a complete data matrix and updates it by relying on both non-missing entries and estimates of missing entries.

It employs a different iterative procedure for finding a factor, which will be referred to as Kiers algorithm and described first. Kiers algorithm operates with a completed version of matrix $\mathbf{X}$ to be denoted by $\mathbf{X}^s$ where $s = 0, 1, ..$ is the iteration's number. At each iteration s, the algorithm finds the best factor of SVD for $\mathbf{X}^s$ and imputes the results into missing entries, after which the next iteration starts.

Kiers Algorithm
1. Set $\mathbf{c}' = (1, \ldots, 1)$ *and normalize it.*
2. Set $s = 0$ *and define matrix* $\mathbf{X}^s$ *by putting zeros into missing entries of* $\mathbf{X}$. *Set a measure of quality* $h_s = \sum_{i=1}^{N} \sum_{k=1}^{n} {x_{ik}^s}^2$.
3. Find the first singular triple $\mathbf{z}_1$, $\mathbf{c}_1$, μ *for matrix* $\mathbf{X}^s$ *by applying the Iterative SVD algorithm with* $p = 1$ *and take the resulting value of criterion (2) as* h_{s+1}.
4. If $|h_s - h_{s+1}| > \epsilon * h_s$ *for a small* $\epsilon > 0$, *set* $s = s + 1$, *put* $z_{i1}c_{1k}$ *for each missing entry* (i, k) *in* $\mathbf{X}$ *and go back to step 3.*
5. Set $\mathbf{z}_1$ *and* $\mathbf{c}_1$ *as the output.*

Now we can formulate IMLS algorithm [10] as follows:

IMLS Algorithm
0. Set the number of factors p.
1. Set iteration number t=1.
2. Apply Kiers algorithm to matrix $\mathbf{X}$ *with the missing structure* $\mathbf{M}$. *Denote results by* $\mathbf{z}_t$ *and* $\mathbf{c}_t$.
3. If $t < p$, *for each* (i, k) *such that* $m_{ik} = 1$, *update* $x_{ik} = x_{ik} - c_{tk}z_{tk}$, *put t=t+1 and go to step 2.*
4. Impute missing values x_{ik} *at* $m_{ik} = 0$ *according to (1) with* $e_{ik} = 0$.

Theoretical properties of the IMLS method remain to be explored.

2.4 Nearest Neighbour-Based Data Imputation

2.4.1 Lazy Learning and Nearest Neighbour

In this section, we are going to apply the so-called lazy learning approach to least-squares methods as described above.

The term "lazy learning" applies to a class of local learning techniques in which all the computation is performed in response to a request for prediction. The request is addressed by consulting data from only a relatively small number of entities considered relevant to the request according to a distance measure [1]. In this framework, the imputations are carried out sequentially, by analyzing entities with missing entries one-by-one. An entity containing one or more of missing entries which are to be imputed is referred to as a target entity. A most popular version of lazy learning is the so-called nearest neighbour (NN) approach (see, for instance, [14]). According to this approach, a distance measure is computed between the target entity and each of the other entities and then K entities nearest to the

target are selected. The imputation model such as (1), for the target entity, is found by using a shortened version of **X** to contain only K+1 elements: the target and K selected neighbours.

Further we consider NN-based versions for the algorithms above.

To apply the NN approach, the following two issues should be addressed.

1. **Measuring distance.** There can be a multitude of distance measures considered. We choose Euclidean distance squared as this measure is compatible with the least-squares framework. The distance between a target entity $\mathbf{X}_i$ and an entity $\mathbf{X}_j$ is defined as:

$$D_2(\mathbf{X_i}, \mathbf{X_j}, \mathbf{M}) = \sum_{k=1}^{n} [x_{ik} - x_{jk}]^2 m_{ik} m_{jk}; i, j = 1, 2, \ldots N \tag{6}$$

   where m_{ik} and m_{jk} are missingness values for x_{ik} and x_{jk}, respectively. This distance was also used in [15,21].
2. **Selection of the neighbourhood.** The principle of selecting the nearest entities can be implemented, first, as is, on the set of all entities, and, second, by considering only entities with non-missing entries in the attribute corresponding to that of the target's missing entry. The second approach was applied in [21] for data imputation with the method Mean. We apply the same approach when using this method. However, for IMLS, the presence of missing entries in the neighbours creates no problems, and, with these methods, we select neighbours among all entities.

2.5 Global–Local Learning Imputation Algorithm

Now, we are ready to introduce the global–local learning of least-squares imputation. This approach involves two stages. First stage: Use a global imputation technique to fill in all the missings in matrix **X**. Let us denote the resulting matrix $\mathbf{X}^*$. Second stage: Apply a lazy learning technique to fill in the missings in **X** again, but, this time, based on distances computed with the completed data $\mathbf{X}^*$. These distances will be referred to as the *prime distances*.

We specify this global–local approach by involving IMLS at both of the stages, which will be referred to as algorithm INI in the remainder. The algorithm INI consists of four main steps. First, impute missing values in the data matrix **X** by using IMLS with $p = 4$. Then compute the prime distance metric with thus found $\mathbf{X}^*$. Take a target entity according to **X** and find its neighbours according to the prime distance. Finally, impute all the missing values in the target entity with NN version of IMLS algorithm (this time, with $p = 1$).

INI Algorithm

1. Apply IMLS algorithm to $\mathbf{X}$ *with* $p = 4$ *to impute all missing entries in matrix* $\mathbf{X}$*; denote resulting matrix by* $\mathbf{X}^*$.
2. Take the first row in $\mathbf{X}$ *that contains a missing entry as the target entity* $\mathbf{X_i}$.
3. Find K neighbours of $\mathbf{X_i}$ *on matrix* $\mathbf{X}^*$.
4. Create a data matrix $\mathbf{X}_c$ *consisting of* $\mathbf{X_i}$ *and rows of* $\mathbf{X}$ *corresponding to the selected K neighbours; the missing pattern is assumed inherited from the original data.*
5. Apply IMLS algorithm with $p = 1$ *to* $\mathbf{X}_c$ *and impute missing values in* $\mathbf{X_i}$.
6. If no missing entries remain, stop; otherwise go back to step 2.

3 EM Algorithm for Imputation of Incomplete Multivariate Normal Data

Maximum-likelihood estimates can often be calculated directly from the incomplete data by specialized numerical methods such as the expectation–maximization (EM) algorithm which was introduced in [2]. Further development of the implementation of EM algorithm for handling missing data was explored in [12,18]. Indeed, the EM algorithm is derived from the old-fashioned idea of handling missing values through iterative steps:

1. Impute the missing values using ad-hoc values.
2. Estimate the parameters of distribution.
3. Re-impute the missing values using the parameters from step 2.
4. Repeat steps 2 and 3 until the iteration converges for pre-specified threshold values.

Formally, the EM algorithm can be illustrated mathematically as follows: suppose the variables and the current estimate of parameter denoted by X and $\theta(t)$, respectively, then the completed-data likelihood, which is composed from missing and observed values, is written as $\ell(\theta|X)$. The E-step of t-th iteration of EM algorithm can be computed as: $\mathscr{Q}(\theta|\theta^t) = \int \ell(\theta|X) f(X_{\text{mis}}|X_{\text{obs}}, \theta = \theta^t) \mathrm{d}X_{\text{mis}}$ where X_{mis}, X_{obs} and f denote the missing values, observed values and probability density function, respectively. The $f(..)$ usually represents multivariate normal distribution. Then θ^{t+1} is chosen as the value of θ which maximize $\mathscr{Q}$. This algorithm has been implemented in [18,20].

Given a complete-data log likelihood from E-step, M-step finds the parameter estimates to maximize the complete-data log likelihood as:

$$\hat{\theta} = SWP[0]N^{-1}E(\mathbf{U}|\mathbf{X}_{\text{obs}}, \theta) \tag{7}$$

The formal approach of EM algorithm can be summarized as follows:

EM Imputation Algorithm

1. Impute the missings values using ad-hoc values.

2. E-Step: Compute the conditional expectation of complete-data log likelihood, U, which is operated as $E(\mathbf{U}|X_{obs}, \theta)$.

3. M-Step: Given complete-data log likelihood from step 2, calculate the parameter estimates $\hat{\theta}$ *from (7).*

4. Set $\theta = \hat{\theta}$*, then repeat steps 2 and 3 until the iteration converges for pre-specified threshold value.*

5. Impute missing values using an appropriate approach based on the found parameters from step 4.

3.1 EM with Different Mechanisms

There are two popular approaches to fill in missing values as shown in step 5 of EM imputation algorithm. In the first approach, the missings are imputed with random values generated from parameters those to be found in the EM computation. This approach is implemented in "Norm" software developed by Schafer which is freely available in [19]. Indeed, this approach is mainly to be implemented within multiple imputation method. In this framework, the missings are imputed more than once using specific simulation. Then, several imputed data sets are analyzed using ordinary statistical techniques (see, for instance, [16–18]).

In either approach, the imputation of missing entries is accomplished under multiple regression scheme using parameters those to be found in the EM computation. This technique is demonstrated by Strauss in [20].

3.2 Multiple Imputation with Markov Chain Monte-Carlo

Multiple imputation method was first implemented in an editing of data survey to create widely public-use data sets to be shared by many end-users. Under this framework, the imputation of missing values is carried out more than once, typically three to ten times, in order to provide valid inferences from imputed values. Thus, MI method is designed mainly for statistical analysis purposes and much attention

has been paid to it in the statistical literature. As described in [17], MI method consists of the following three-step process:

1. Imputation: Generate m sets of reasonable values for missing entries. Each of these sets of values can be used to impute the unobserved values. Thus, there are m "completed" data sets. This is the most critical step since it is designed to account for the relationships between unobserved and observed variables. Thus the missing at random (MAR) assumption is the central issue to the validity of the multiple imputation approach. There are a number of imputation models that can be applied. Probably the imputation model via the Markov Chain Monte-Carlo (MCMC) is the most popular approach. This simulation approach is demonstrated within the following IP (Imputation-Parameter steps) algorithm [18]:

   I-step: Generate $\mathbf{X}^{\text{mis},t+1}$ from $f(\mathbf{X}|\mathbf{X}^{\text{obs}}, \theta^t)$.
   P-step: Generate θ^{t+1} from $f(\theta|\mathbf{X}^{\text{obs}}, \mathbf{X}^{\text{mis},t+1})$.

   The above steps produce Markov chain $(\{\mathbf{X}^1, \theta^1\}, \{\mathbf{X}^2, \theta^2\}, \ldots, \{\mathbf{X}^{t+1}, \theta^{t+1}\}, \ldots)$ which converge to the posterior distribution.
2. Analysis: Apply the ordinary statistical method to analyze each "completed" data sets. From each analysis, one must first calculate and save the estimates and standard errors. Suppose that $\widehat{\theta_j}$ is an estimate of a scalar quantity of interest (e.g. a regression coefficient) obtained from data set j $(j = 1, 2, \ldots, m)$ and ${\sigma_{\widehat{\theta,j}}}^2$ is the variance associated with $\widehat{\theta_j}$.
3. Combine the results of analysis.
   In this step, the results are combined to compute the estimates of the within-imputation and between-imputation variability [16]. The overall estimate is the average of the individual estimates:

$$\bar{\theta} = 1/m \sum_{j=1}^{m} \theta_j \tag{8}$$

   For the overall variance, one must first calculate the within-imputation variance:

$${\bar{\sigma_\theta}}^2 = 1/m \sum_{j=1}^{m} \sigma^2_{\widehat{\theta,j}} \tag{9}$$

   and the between-imputation variance:

$$B = 1/(m-1) \sum_{j=1}^{m} (\widehat{\theta_j} - \bar{\theta})^2 \tag{10}$$

then the total variance is:

$$\sigma^2_{\text{pool}} = \bar{\sigma_\theta}^2 + (1 + 1/m)B \tag{11}$$

Thus, the overall standard error is the square root of σ^2_{pool}. Confidence intervals are found as: $\bar{\theta} \pm \sigma_{\text{pool}}$ with degrees of freedom:

$$\text{df} = (m-1)\left(1 + \frac{m\bar{\theta}}{(m+1)B}\right) \tag{12}$$

In the context of data imputation, in our view, MI can be applied to estimate missing data as average, estimates of the multiple imputations.

4 Experimental Setting

In this experiments, the benchmarking of global–local least-squares imputation, INI, and two versions of EM imputation as described in previous section will be evaluated. The comparison is accomplished on number samples generated from one real large-scale database.

4.1 Selection of Algorithms

The following three algorithms are selected to expose the experimental study

1. INI: the global–local versions of least-squares imputation [22, 23].
2. EM-Strauss: EM algorithm with multiple regression imputation [3, 20].
3. EM-Schafer: EM algorithm with with random imputation as implemented in [3, 18].
4. MI: Multiple imputation with Markov-Chain Monte Carlo simulation using ten imputations [3, 19]

4.2 Description of Data Set

The data set is produced from the real-world marketing research activity. This original data set consists of 5,001 entities and 65 attributes which consist of mostly numeric (60), categorical (2) and binary attributes (3).

4.3 Generation of Missings

The missings are generated randomly on the original real data set (size 5001×65) at three levels of missings: 1, 5 and 10 %.

4.4 Samples Generation

This experiment utilizes 50 samples (size:250×20) which generated randomly from original database for each missing generation.

4.5 Data Pre-processing

Within the imputation techniques while the experiment running, the data pre-processing, especially for real data sets, is calculated in the following procedures:

$$x_{ik} = \frac{(x_{ik} - \mu_k)}{\text{range}_k} \tag{13}$$

where μ_k and range_k defined as mean and range of attribute, respectively, and they are calculated as:

$$\mu_k = \frac{\sum_{i=1}^{N} x_{ik} * m_{ik}}{N} \tag{14}$$

$$\text{range}_k = \max_i(x_k) - \min_i(x_k) \tag{15}$$

4.6 Evaluation of Results and Performance

Since the data and missings are generated separately, we can evaluate the quality of imputation by comparing the imputed values with those generated at the stage of data generating. We use the squared imputation error, *IE*, to measure the performance of an algorithm. The measure is defined as follows:

$$\text{IE} = \frac{\sum_{i=1}^{N}\sum_{k=1}^{n}(1 - m_{ik})(x_{ik} - x_{ik}^{*})^2}{\sum_{i=1}^{N}\sum_{k=1}^{n}(1 - m_{ik})x_{ik}^2} \tag{16}$$

Table 1 The comparative results of the performances of three algorithms according to their occurrences within specified ranges of error of imputation and their average CPU time (in seconds) where 1, 2, 3 and $\hat{\mu}$ denote INI, EM-Strauss, EM-Schafer and average in interval, respectively

		1%			5%			10%		
Error (%)	$\hat{\mu}$	1	2	3	1	2	3	1	2	3
≤ 35	25	11	9	3	5	10	3	6	6	1
		(1.23)	(54.82)	(0.04)	(2.34)	(292.76)	(0.72)	(2.94)	(1,293.60)	(1.45)
≤ 100	70	27	9	10	38	22	13	37	24	16
		(0.96)	(28.94)	(0.17)	(2.22)	(310.91)	(0.33)	(3.03)	(1,856.80)	(0.81)
$\leq 1{,}000$	750	9	25	18	6	15	22	6	17	21
		(0.87)	(36.23)	(0.28)	(2.10)	(0.61)	(379.65)	(3.98)	(1,502.60)	(0.70)
$> 1{,}000$	1,300	3	7	18	1	3	7	0	2	5
		(0.96)	(31.33)	(0.11)	(1.65)	(2,082.2)	(0.25)	(–)	(1,185.30)	(0.25)
NaN	–	0	0	1	0	0	5	1	1	7
		(–)	(–)	(–)	(–)	(–)	(–)	(–)	(–)	(–)

where m_{ik} is the missingness matrix entry and x^*_{ik} an entry in the data matrix $\mathbf{X}^*$ with imputed values. To evaluate the performance of the imputation methods, the elapsed CPU time for running the program at Pentium III 733 MHz is recorded.

4.7 Results

The experiments are carried out in two settings: (1) Experiments involving INI and two EM imputation versions: EM-Strauss and EM-Schafer. In these experiments, 50 samples are used for each level of missings. Thus there are 150 samples in the experiments; (2) Experiments involving INI, EM-Strauss, EM-Schafer and multiple imputation with ten times imputation for each data sample. In these experiments, 20 samples are used for two levels of missings: 5 and 10 %. The results of each experiment will be shown in turn.

4.7.1 The Experiments with INI and Two EM Imputation Versions

The results of series experiments are summarized in Table 1. The error of imputation is classified into five groups including in case the algorithms cannot be proceeded which is labeled as "NaN". The failing of computation is caused by the nature of algorithm. It can be described as follows. INI cannot be implemented in case the subset of data matrix that to be found by k-NN algorithm contains all zeros elements in one column or more. Thus, the Eqs. (4) and (5) cannot be computed. Finally, the imputed values cannot be found. On the other hand, for both versions of EM algorithm, the full covariance matrix that to be found from EM computation should be positive definite; otherwise, imputed values cannot be calculated.

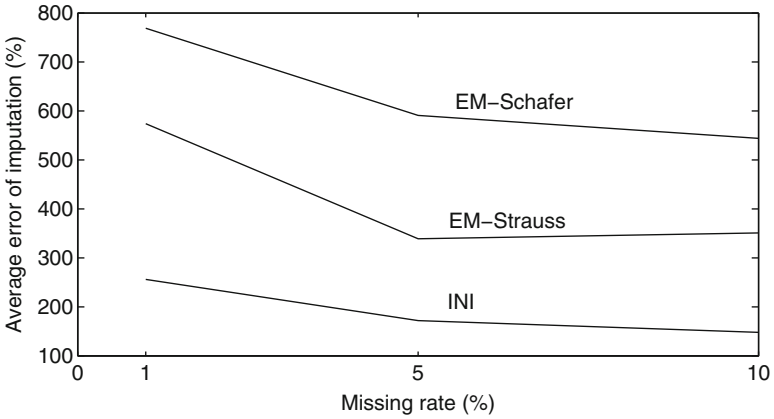


Fig. 1 The average error of imputation

According to Table 1, at level 1 % missings, for error of imputation is 100 % or less, INI surpasses both EM algorithms. Furthermore, followed by EM-Strauss as second winner and the EM-Schafer to be the worst. On the other hand, regarding CPU time performance, EM-Schafer algorithm provides the most fastest of rate of convergence and INI to be the second fastest.

As the level of missings increased to 5 %, for 100 % or less of error of imputation, INI, still, surpasses both EM-Strauss and EM-Schafer. However, at level 35 % or less of error of imputation, EM-Strauss beats INI. According to CPU time measurement, the EM-Strauss produces the most slowest rate of convergence. Thus, overall, INI still surpasses the other methods.

Finally, at level of 10 % missings, for 100 % or less of error of imputation, again, INI surpasses the EM-Strauss and EM Schafer. Furthermore, at level 35 % or less of error of imputation, INI and EM-Strauss provide the same occurrences and the EM-Schafer consistently to be the worst. In contrast, according to the CPU time measurement, EM-Schafer consistently to be the fastest method.

Figure 1 shows how average error changes. It is clear that INI outperforms two others for all situations. Figure 2 shows also that standard deviation of INI is much smaller than that for others two. Figures 3, 4, and 5 show the performance of algorithms for three levels of missing values. It is seen that INI's distribution is, first, much sharper (the mode is more than 75 % twice and more than 50 % once versus 35–50 % for others), and, second, is much more consistent in shape than that of two other methods. It tells, in general, about better stability of the approach when data is changed.

Table 2 shows the coefficient of variation for INI is always larger than for two others, which is to be explained by the fact that smaller level of average error (in denominator) is not surrounded with respected decrease of standard deviation (the share of high fluctuations in zone of very large errors, $> 1{,}000$ %, is more significant for smaller levels).

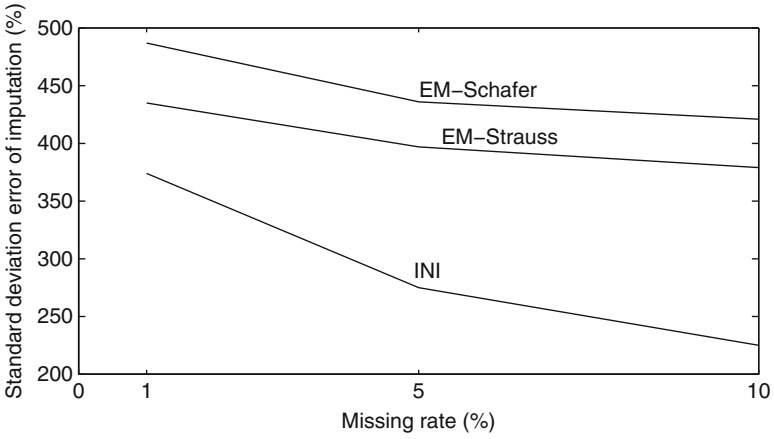


Fig. 2 The standard deviation error of imputation

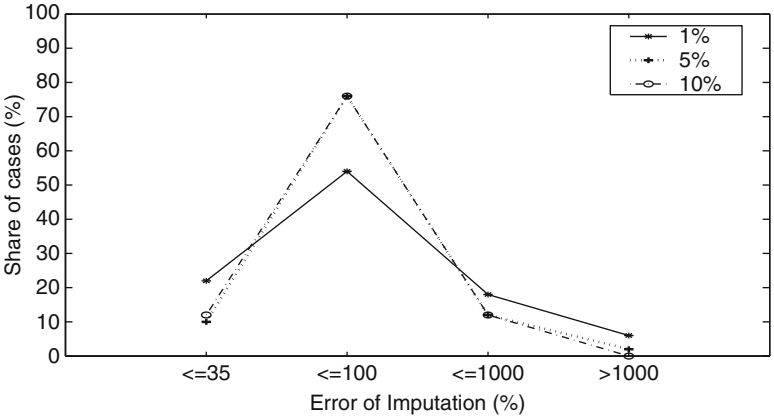


Fig. 3 The performance of INI algorithm

4.7.2 The Experiments with INI, Two EM Imputation Versions and MI

This time, the experiments are carried out using 20 samples out of 50 samples which are used in the previous experiments. The samples are chosen from "population" with level of missings: 5 and 10 %. The error of imputation for each method is presented in Table 3.

The result of experiment is summarized according to the pair-wise comparison of imputation methods: INI, EM-Strauss, EM-Schafer and MI with ten times imputation for each sample. The comparison is shown in Table 4.

Table 4 shows that at level 5 %, three methods, INI, EM-Strauss and MI, provide almost the similar results. However, in the close range, EM-Strauss appears as the

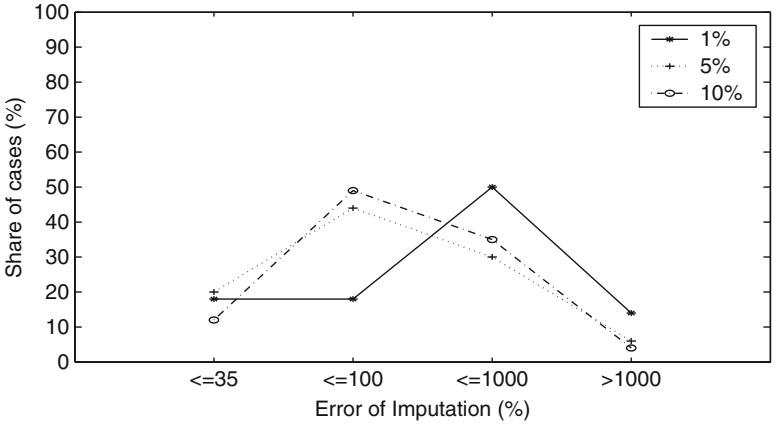


Fig. 4 The performance of EM-Strauss algorithm

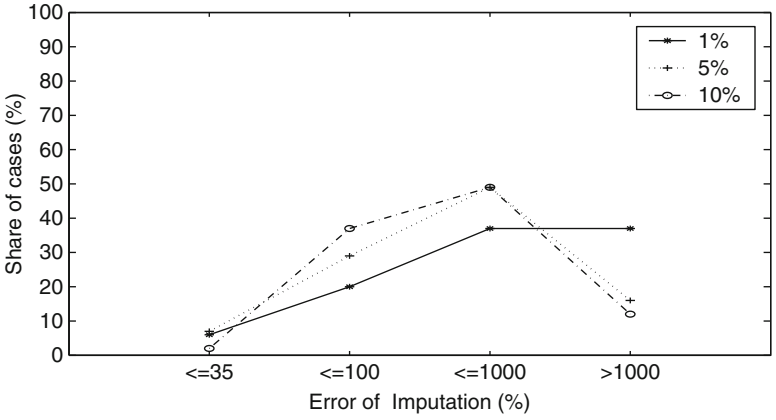


Fig. 5 The performance of EM-Schafer algorithm

Table 2 The various statistical measurement of the performances of three algorithms where 1, 2 and 3 denote INI, EM-Strauss, EM-Schafer, respectively

	1%			5%			10%		
Stats	1	2	3	1	2	3	1	2	3
Average	256	574	769	172	339	591	148	351	544
Standard deviation	374	435	487	275	397	436	225	379	421
Coefficient of variance (%)	146	76	63	160	117	74	153	108	77

best method. Then MI appears as the second best. However, as the level of missings increases to 10 %, INI surpasses the other methods. Then it is followed by EM-Strauss. As shown in the previous experiments, the EM-Schafer consistently to be the worst method.

Table 3 The squared error of imputation (in %) of INI, EM-Strauss, EM-Schafer and MI on 20 samples at 10 % missings entry where NN denotes the methods fail to proceed

	Methods			
Samples	INI	EM-Strauss	EM-Schafer	MI
1	73.78	91.35	NN	28.32
2	95.98	835.21	575.87	24.45
3	57.78	53.89	58.21	545.72
4	43.68	45.10	73.88	129.34
5	NN	NN	NN	40.99
6	48.35	59.94	58.20	144.32
7	61.28	51.40	89.86	99.91
8	142.80	307.59	1,048.37	95.52
9	97.29	86.93	128.11	126.62
10	53.79	56.70	109.95	50.52
11	73.56	92.00	235.75	NN
12	75.86	293.90	184.65	389.28
13	134.05	840.37	5,429.77	57.07
14	62.17	41.53	136.28	49.51
15	78.97	360.20	NN	NN
16	67.80	113.21	723.93	57.63
17	44.93	63.34	62.96	50.76
18	74.37	71.53	NN	333.34
19	72.44	78.21	150.24	87.83
20	78.68	115.89	542.38	51.86

Table 4 The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on 20 samples generated from database with 5 and 10 % random missing data where 1,2,3 and 4 denote INI, EM-Strauss, EM-Schafer and MI, respectively

	5%				10%			
Methods of imputation	1	2	3	4	1	2	3	4
INI	–	50	30	55	–	26	0	47
EM-Strauss	50	–	25	45	74	–	25	47
EM-Schafer	70	75	–	80	100	75	–	67
MI	45	55	20	–	53	53	33	–

5 Conclusion

We described a number of least-squares data imputation techniques. These methods extend the one-by-one extraction strategy of the principal component analysis to the case of incomplete data and combine it with the nearest neighbour approach as proposed in [22, 23]. We also reviewed expectation–maximization (EM)-based approach and multiple imputation for handling missing data as described in [18,20].

We carried out experimental comparisons on marketing research data within simulation framework. It appears, overall, the global–local two-stage NN-based method INI overwhelmingly outperforms EM-based methods and is comparable with multiple imputation (MI) approach.

Acknowledgements The author gratefully acknowledges many comments by reviewers that have been very helpful in improving the presentation.

References

1. Aha, D.:Editorial. Artif. Intel. Rev. **11**, 1–6 (1997)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. **39**, 1–38 (1977)
3. EM Based Imputation Software. http://www.stat.psu.edu/jls/misoftwa.html, http://methcenter.psu.edu/EMCOV.html (1995)
4. Gabriel, K.R., Zamir, S.: Lower rank approximation of matrices by least squares with any choices of weights. Technometrics **21**, 489–298 (1979)
5. Golub, G.H., Loan, C.F.: Matrix Computation, 2nd edn. John Hopkins University Press, Baltimore (1986)
6. Heiser, W.J.: Convergent computation by iterative majorization: theory and applications in multidimensional analysis, In: Krzanowski, W.J. (ed.) Recent Advances in Descriptive Multivariate Analysis, pp. 157–189. Oxford University Press, Oxford (1995)
7. Ho, Y., Chung, Y., Lau, K.: Unfolding large-scale marketing data. Int. J. Res. Mark. **27**, 119–132 (2010)
8. Holzinger, K.J., Harman, H.H.: Factor Analysis. University of Chicago Press, Chicago (1941)
9. Jollife, I.T.: Principal Component Analysis. Springer, New-York (1986)
10. Kiers, H.A.L.: Weighted least squares fitting using ordinary least squares algorithms. Psychometrika **62**, 251–266 (1997)
11. Laaksonen, S.: Regression-based nearest neighbour hot decking. Comput. Stat. **15**, 65–71 (2000)
12. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley, New York (1987)
13. Mirkin, B.: Mathematical Classification and Clustering. Kluwer Academic, Dordrecht (1996)
14. Mitchell, T.M.: Machine Learning. McGraw-Hill, London (1997)
15. Myrtveit, I., Stensrud, E., Olsson, U.H.: Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods. IEEE Trans. Softw. Eng. **27**, 999–1013 (2001)
16. Rubin, D.B.: Multiple Imputation for Nonresponse in Surveys. Wiley, New York (1987)
17. Rubin, D.B.: Multiple imputation after 18+ years. J. Am. Stat. Assoc. **91**, 473–489 (1996)
18. Schafer, J.L.: Analysis of Incomplete Multivariate Data. Chapman and Hall, London (1997)
19. Schafer, J.L.: NORM. http://www.stst.psu.edu/jls/misoftwa.html (1997)
20. Strauss, R.E., Atanassov, M.N., De Oliveira, J.A.: Evaluation of the principal-component and expectation-maximization methods for estimating missing data in morphometric studies. J. Vertebr. Paleontol. **23**(2), 284–296 (2003)
21. Troyanskaya, O., Cantor, M. , Sherlock, G., Brown, P., Hastie, T., Hastie, R., Tibshirani, R., Botsein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics **17**, 520–525 (2001)
22. Wasito, I., Mirkin, B.: Nearest neighbour approach in the least-squares data imputation algorithms. Inf. Sci. **169**, 1–25 (2005)
23. Wasito, I., Mirkin, B.: Least squares data imputation with nearest neighbour approach with different missing patterns. Comput. Stat. Data Anal. **50**, 926–949 (2006)

AST Method for Scoring String-to-text Similarity

Ekaterina Chernyak and Boris Mirkin

Abstract A suffix-tree-based method for measuring similarity of a key phrase to an unstructured text is proposed. The measure involves less computation and it does not depend on the length of the text or the key phrase. This applies to:

1. finding interrelations between key phrases over a set of texts;
2. annotating a research article by topics from a taxonomy of the domain;
3. clustering relevant topics and mapping clusters on a domain taxonomy.

Keywords Suffix tree • Unstructured text analysis • String similarity measures

1 Introduction

Typically, string-to-text similarity measures are defined using the vector space model (VSM) text representing model. Here, a richer text model, the suffix tree, is used to keep the sequential nature of sentences and make text analysis independent from the natural language and its grammar [3, 9, 12]. Conventional suffix-tree-based similarity measures also suffer from drawbacks related to the intensity of computations and their sensitivity to the lengths of both texts and strings. We propose a measure that allows relaxing these limitations. Then we apply our similarity measure to two types of problems:

1. Analysis of interrelations between key phrases over a text collection;
2. Annotation of research articles by a corresponding domain taxonomy topics;

E. Chernyak (✉) • B. Mirkin
School of Applied Mathematics and Information Science, National Research University – Higher School of Economics, Moscow, Russia
e-mail: echernyak@hse.ru; bmirkin@hse.ru

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_20,

3. Analysis of teaching syllabuses and resident complaints by mapping them to taxonomies, clustering the taxonomy topics, and lifting the clusters over the taxonomy tree.

These three problems may look close to traditional information retrieval task as stated in [10, 11], where the main task is to find all relevant documents for the given query or even to rank them according to their relevance to the query. The key difference is that:

1. We fix the set of queries (e.g., key phrases or taxonomy). What is more, in problem of type 1 we investigate the structure of this set. In [10] no queries are looked at, expect the given one, which is the matter of document ranking problem.
2. We consider all the words occurred in the texts, not only those, which appear in the queries as it is done in [10, 11].

Hence we treat the set of queries, Q, and the set of txt or documents, D, as a two fixed sets of words, further combined in the so-called strings, of equivalent importance for the analysis.

Section 2 describes our method for an annotated suffix tree (AST) construction and the scoring function. Section 3 briefly explains the basic concept of the ST table used throughout in computations. Section 4 presents methods for solving a problem of type (1). Section 5 applies this to a problem of type (2). Two problems of type (3) are described in Section 6. The conclusion completes the text.

2 AST method

The suffix tree is a data structure used for storing of and searching for symbolic strings and their fragments [4]. In a sense, the suffix tree model is an alternative to the VSM, arguably, the most popular model for text representation [12]. When the suffix tree representation is used, the text is considered as a set of strings, where a string may be any semantically significant part of the text, like a word (like it is done in the Bag-of-Words model), a phrase, or even a whole sentence. An AST is a suffix tree whose nodes (not edges!) are annotated by the frequencies of the strings fragments.

We split texts into short fragments, “strings”, to reduce the computation. Usually we take the strings to be of three sequential words [2]. An AST for a string is a rooted tree, in which each node is labeled with one of the string symbols so that each path from the root to a leaf encodes one of the string suffixes. AST for a set of strings stores all the fragments of all the strings and their frequencies (see Fig. 1). Using AST representation of texts, one is able to find the most frequent fragments in the text and their length.

To build an AST for a text, for its every string, its suffixes are added to the AST, starting from an empty set. To add a suffix to the AST, first check whether there is

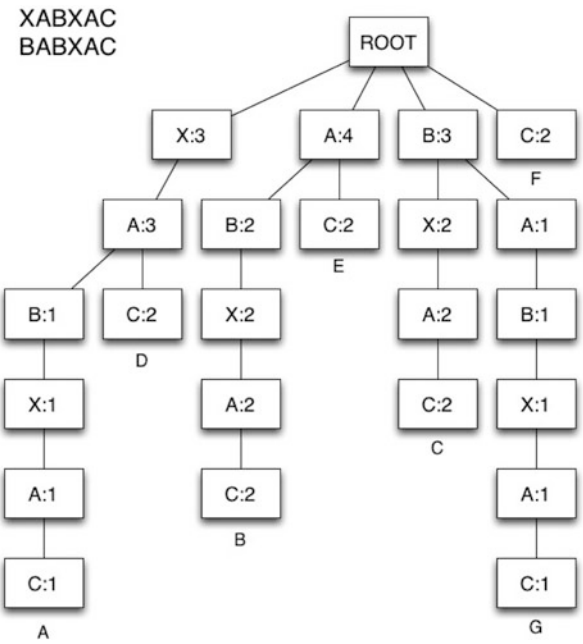


Fig. 1 AST for two strings XABXAX and BABXAC. Note that the strings differ only in the first position and have common five suffixes

already a path in the AST that encodes the whole suffix or its prefix. If such a path (a match) exists, we increase all the frequencies in the match and append new nodes with frequencies 1 to the last node in the match, if it doesn't cover the whole suffix. If there is no match, we create a new chain of nodes in the AST with frequencies 1.

To score similarity of a string to an AST, we match all its fragments with the AST and score each match as the sum of conditional probabilities of matching nodes divided by the length of the match. The conditional probability is the ratio of the node frequency to that of its parent. If no match is found, define the zero score. Then the average of all the scores is computed:

$$\begin{aligned} \text{scoreMatch}(\text{string}, \text{AST}) &= \frac{\sum_{\text{suffix}} \text{score}(\text{suffix}, \text{AST})}{\text{length}(\text{string})} \\ &= \frac{\sum_{\text{suffix}} \frac{\sum_{u \in \text{match}} f_u / f_{\text{parent}(u)}}{\text{length}(\text{match})}}{\text{length}(\text{string})} \end{aligned} \tag{1}$$

The VSM-based models are based on finding word-to-word exact coincidence: a phrase, which is a set of words, may be considered relevant to a text, if a significant part of that set occurs in the text. The AST measure avoids searching for exact occurrences. It takes all matching fragments into account, so that a word may match two or more times to the text. Hence, when estimating string-to-text similarity, we deal not with the space of words, but with the space of different matching fragments.

3 Building an ST Table

To analyze the relationship between a set of strings and a collection of texts, we build a string-to-text similarity table (ST table) by constructing an AST for each of the texts and estimating similarity of each of the strings to this AST. The rows of ST table correspond to the strings and columns, to the texts. Using an ST table allows us to treat strings as numerical attributes and exploit thus conventional data analysis techniques.

4 Analysis of Interrelations on a Set of Strings over a Related Text Collection

Consider a collection of web publications about current business processes in Russia and a set of key phrases that describe local events like "publishing financial reports" or "replacement of the finance management." To find relations between these events, an ST table key_phrase–to–web_ publication is built first. There can be three types of web publications:

1. those related to only one key phrase;
2. those related to two or more key phrases;
3. those related to no key phrases.

By specifying a threshold, we assign each of the key phrases A with a subset F(A) of related web publications.

Key phrase A implies B, if proportion of F(B) in F(A) is greater than 60 %. Thus, one can draw a graph of the implications. For example, in the analysis of 960 web publications on business processes in 2009 with about 40 key phrases, we discovered that only 12 of them are of type (2) (see Fig. 2).

5 Annotation of Journal Articles by Topics from a Taxonomy of the Domain

Another application of the AST method is indexing scientific papers with topics of a taxonomy of the domain. Consider the association of computing machinery (ACM) journals and the ACM developed taxonomy "Computing Classification System" [1] (ACM-CCS). To represent the contents of their papers, authors of the ACM journals annotate the papers manually with topics from the ACM-CCS taxonomy. To automate this procedure using the AST method one has to:

1. Extract key elements of a paper, such as its heading, abstract, keywords if given.
2. Build an AST for the extracted elements.

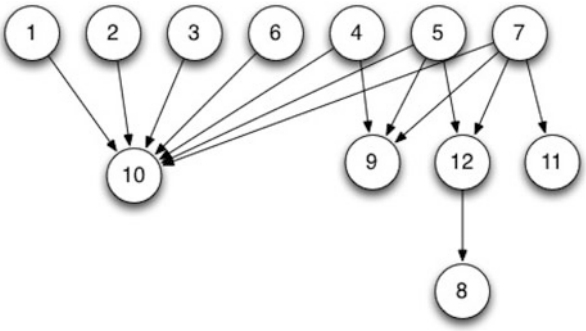


Fig. 2 Graph of the interrelation between the key phrases over a text collection. Codes: 1: Introduction of automated manufacturing; 2: Issuing news bulletins; 3: Change of the size of the shares belonging to the institutional investor; 4: Change of the extent of the ownership concentration; 5: Personnel training; 6: Vertical merger; 7: Brand selling/buying; 8: Entering international markets; 9: Change of the legal organizational form; 10: More effective cost control; 11: Making the finance reports publicly available; 12: Change of the finance director

Table 1 Profile A

Bojanczyk M., et al., Two-variable logic on data trees and XML reasoning
Journal of the ACM, 2009, Vol. 56(3), pp. 2–48

AST found profile			ACM-CCS index terms (manual annotation)		
ID	S	ACM-CCS topic	ID	Rank	ACM-CCS topic
I.6.2	0.4969	Simulation languages	F.4.3	3	Formal languages
I.1.3	0.4415	Languages and systems	H.2.3	4	Languages
F.4.3	0.3796	Formal languages	H.2.1	13	Logical design
H.2.3	0.3757	Languages	F.4.1	28	Mathematical logic
D.4.5	0.2738	Reliability	I.7.2	53	Document preparation

3. Estimate the similarity of every ACM-CCS topic to the text. The topic similarity values form what we refer to as the AST-profile of the publication.
4. Choose the ACM-CCS topics with the highest scores.

There are two examples of the so-called ACM abstract profiles. We put on the left side of each profile the top five taxonomy topics, sorted according to taxonomy topic to abstract similarity measure. The manual annotation chosen by authors is on the right side. The ID stands for the topic ID in the ACM-CCS taxonomy, S is the similarity value, ACM-CCS topic is the topic itself, Rank is the place where the manual annotation has been placed in the AST profile. Ranks can help us to estimate the quality of profiles: the higher ranks manual annotations get, the better the profile is. Hence the Profile A can be thought of as a rather good one and the Profile B as a poor one (Tables 1 and 2).

Unfortunately, AST profiles are not always close to the manual annotations. This may happen because:

Table 2 Profile B

Grohe M., et al., Lower bounds for processing data with few random accesses to external memory Journal of the ACM, 2009, Vol. 56(3), pp. 1–58

AST found profile			ACM-CCS index terms (manual annotation)		
ID	S	ACM-CCS topic	ID	Rank	ACM-CCS topic
J.1	0.5991	Administrative data processing	F.1.3	161	Complexity measures and classes
I.2.7	0.4757	Natural language processing	H.2.4	166	Systems
H.2.5	0.4704	Heterogeneous databases	F.1.1	220	Models of computation
H.2.8	0.3419	Database applications			
C.5.1	0.3146	Large and medium computers			

A. The method evaluates common words, such as "theorem," "method," or problem" too high. This issue can be addressed by using a stop list of common words.
B. The method works when the formulations of topics use similar letters. It doesn't cope with synonyms. A solution to this issue would be taking into account a set of synonyms and near synonyms for each of the taxonomy topics.
C. The authors sometimes go too far in their annotations by assuming implications of their methods which are not much considered in the text.

6 Clustering Relevant Topics and Mapping Clusters on a Domain Taxonomy

Suppose we have a collection of texts and a taxonomy, which belong to the same topic domain. We treat taxonomy as a set of topics, each presented by only one string, organized in a rooted tree. The higher the topic is, the more general it is. Hence we can employ the AST method to construct the ST table. In such a table rows, i.e. strings, stand for leaf taxonomy topics, and columns for the texts. Note that we restrict ourselves only to leaf topics, because it is essential for further analysis. However all the topics might be used in the way as it is described in the previous section. According to the AST table we may find groups of similar topic which match with the text in almost the same fashion by means of some cluster analysis methods. First of all, the clusters may be of their own interest, because they consist of topics that appear in texts similarly although they don't necessarily belong to one branch of taxonomy. Secondly, using the lifting method, we can map these clusters into the taxonomy. The lifting method outputs a few taxonomy topics of higher levels which cover the cluster of leaf topics in the best possible way. This allows to interpret the whole collection of texts in terms of several taxonomy topics of higher

levels, that is a way of data aggregation over hierarchically organized taxonomy. Let us enumerate the main steps of the cluster-lift method:

1. constructing the leaf_taxonomy_topic text ST table
2. finding clusters of leaf taxonomy topics
3. mapping the clusters into higher levels of the taxonomy structure.

To cluster the ST table we may first find leaf_taxonomy_topic leaf_taxonomy_topic similarity matrix by taking dot products of rows of the ST and apply then Additive Fuzzy Spectral Clustering (FADDIS) method [6,7] that uses the Spectral Clustering approach to the Additive Fuzzy Clustering Model to find clusters of leaf taxonomy topics. The other possible way to cluster leaf taxonomy topics is to use iK-Means [5] method to extract clusters one by one from the ST table. The final step is to "lift" the clusters in the taxonomy. The lifting algorithm [7] proceeds according to the assumption that if all or almost all elements of a cluster could be covered by a topic on a higher levels than the whole cluster "lifts" to that very topic. If the assumption does not hold, then lifting is impossible. More details on all the methods used are provided in [6]. Below two applications of the cluster-lift method are presented.

6.1 Teaching Syllabuses and the Taxonomy of Mathematics and Informatics

The input is twofold. First, we take the most extensive taxonomy of mathematics and informatics domain in Russian [8], that is called the VINITI taxonomy. It is an unbalanced and rather messy rooted tree of mathematics and informatics topics, provided with a lot of cross-references. Second, we downloaded from the web page (www.hse.ru) of our university a collection of teaching syllabuses. These syllabuses correspond to all courses related to Mathematics and/or Informatics as they are taught in the School of Applied Mathematics and Informatics of NRU HSE. The study of the VINITI taxonomy and the collection of teaching syllabuses shows several shortcomings, both of the syllabuses and of the taxonomy: almost every cluster we get after applying the method to the data contained topics from the Topology branch of the taxonomy. It means that one or another notion form topology is studied during almost all mathematical courses. But there is no such a subject in the curriculum.

As the VINITI taxonomy has not been updated since the early 1980s, it was expected that it may have issues in covering more modern topics in mathematics and informatics. With the help of teaching syllabuses we establish several nests of topics that should be possibly added to the ontology. For example, the topic "Lattices" is by now a leaf in the taxonomy. According to our results, it should be a parent node with three offsprings: "Modular lattices", "Distributive lattices," and "Semimodular lattices."

The taxonomy has been found of rather imbalanced in the coverage. The "Differential Equations" and "Mathematical Analysis" branches are significantly more saturated than the other branches and comprise almost the half of the taxonomy. Yet less classical branches as "Game Theory" or "Programming Theory" are way too small and not comprehensive at all. They build up a very small part of the taxonomy, especially in comparison with the giant branches "Differential Equations" and "Mathematical Analysis". We thought that the main teaching syllabuses should be named after the first-level or second-level taxonomy topics. On the contrary, we found that such topics as "Discrete Mathematics" have not been set among the high-layer taxonomy topics in the VINITI taxonomy.

In this study we used the FADDIS model to cluster the leaf taxonomy clusters. Unfortunately, because of cluster elements being from disconnected branches of taxonomy, the lifting procedure almost failed. We only achieved one layer up lift in a few situations, such as "Continuous distributions" and "Discrete distributions" being lifted to their common parent "Probability distributions".

6.2 Resident Complaints and the Taxonomy of Community Facilities

During the last years special systems for submitting any kind of complaints from residents are introduced in some cities in Russia. One of those systems is exploited in Nizhniy Novgorod for the residents complaining on the problems with community facilities. Our colleagues from the Nizhniy Novgorod NRU HSE campus developed a taxonomy that describes almost all constituents of community facilities in order to automate the analysis of the flow of complaints.

First we noticed that there are many significant concepts missing from this taxonomy. For example, there were topics about "Elementary school" and "Middle school", while the "Kindergarten" topic was missing. We manually extracted some frequent nouns or collocations from the given collection of complaints and updated the taxonomy with these extracted topics.

Second we built the leaf_taxonomy_topic resident complaint ST table by means of the AST method and then applied the cluster-lift method. We used iK-means method [5] to extract clusters of taxonomy topics from the ST table. We cleaned the clusters from extra large or small clusters. The rest of the clusters were parsimoniously lifted. For example, cluster 1 in Fig. 3 consisted of four taxonomy topics: 1.2.1. Hot water problems, 1.2.2 Cold water problems, 1.2.3 Water meter problems, 1.11.2 Public water pump. On the first iteration of the lifting method the cluster was mapped to 1.2 Water Supply and to 1.11 Urban landscaping and public amenities. These two were then mapped on the second iteration to the first level topic 1. Housing services.

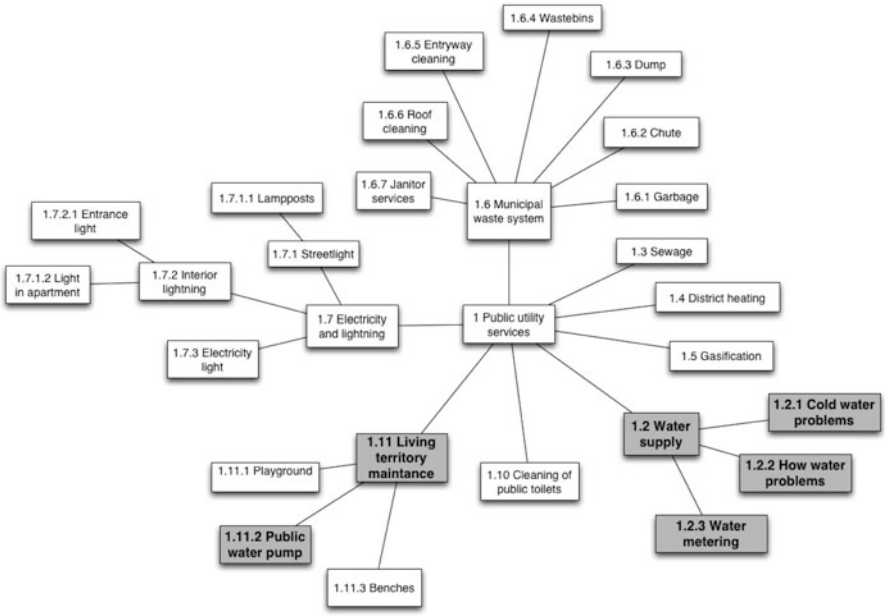


Fig. 3 Lifting cluster to higher levels

7 Conclusion

The AST method used for estimating string-to-text similarity has several advantages over the VSM-based methods. It doesn't require any complicated preprocessing procedure like stemming or POS-tagging and is then independent from grammar. By using the string concept, we can explore long enough text fragments, so that short semantical links aren't lost. The experimental computations lead us to the number of issues that are to be subject of the further developments:

1. the AST method deals only with matching strings. To make it more efficient we should take synonyms or near synonyms into account.
2. we have done so far some manual attempts to improve taxonomies so that they would become more balanced and up to date. It is, perhaps, possible to automate the process of improving or refining taxonomies using either given collections of texts or some external sources.
3. by now we have assumed that the set of strings that is further used for matching with the texts and building the ST table is given by some experts or is taken from some official source. We may extract these strings from the texts by using some well-known methods of long key phrases extraction.

References

1. ACM Computing Classification System. http://www.acm.org/about/class/ (1998)
2. Chernyak, E., Chugunova, O., Askarova, J., Nascimento, S., Mirkin, B.: Abstracting concepts from text documents by using an ontology. In: 1st International Workshop on Concept Discovery in Unstructured Data, pp. 20–30. University Higher School of Economics, Moscow (2011)
3. Grossi, R., Vitter, J.: Compressed suffix arrays and suffix trees with applications to text indexing and string matching. SIAM J. Comput. **35**(2), 378–407 (2005)
4. Gusfield, D.: Algorithms on Strings, Trees, and Sequences. Cambridge University Press, Cambridge (1997)
5. Mirkin, B.: Clustering for Data Mining: A Data Recovery Approach. Chapman and Hall/CRC, Boca Raton (2005)
6. Mirkin, B., Fenner, T., Nascimento, S., Pereira, L.M.: A Hybrid cluster-lift method for the analysis of research activities. Lect. Notes Comput. Sci. **6076**(1), 152–161 (2010)
7. Mirkin, B., Nascimento, S., Fenner, T., Pereira, L.M.: Fuzzy thematic clusters mapped to higher ranks in a taxonomy. Int. J. Softw. Inform. **4**(3), 257–275 (2010)
8. Nikol'skaya, I.Y., Yefremenkova, V.M.: Mathematics in VINITI RAS: from abstract journal to databases. Sci. Tech. Inf. Process. **35**(3), 128–138 (2008) (in Russian)
9. Pampapathi, R., Mirkin, B., Levene, M.: A suffix tree approach to anti-spam email filtering. Mach. Learn. **65**(1), 309–338 (2006)
10. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. J. Found. Trends Inf. Retr. **3**(4), 333–369 (2009)
11. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM. **18**(11), 613–620 (1975)
12. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: Proceedings of SIGIR'98, pp. 46–54. University of Washington, Seattle (1998)

Improving Web Search Relevance with Learning Structure of Domain Concepts

Boris A. Galitsky and Boris Kovalerchuk

Abstract This paper addresses the problem of improving the relevance of a search engine results in a vertical domain. The proposed algorithm is built on a structured taxonomy of keywords. The taxonomy construction process starts from the seed terms (keywords) and mines the available source domains for new terms associated with these entities. These new terms are formed in several steps. First the snippets of answers generated by the search engine are parsed producing parsing trees. Then commonalities of these parsing trees are found by using a machine learning algorithm. These commonality expressions then form new keywords as parameters of existing keywords and are turned into new seeds at the next learning iteration. To match NL expressions between source and target domains, the proposed algorithm uses syntactic generalization, an operation which finds a set of maximal common sub-trees of constituency parse trees of these expressions. The evaluation study of the proposed method revealed the improvement of search relevance in vertical and horizontal domains. It had shown significant contribution of the learned taxonomy in a vertical domain and a noticeable contribution of a hybrid system (that combines of taxonomy and syntactic generalization) in the horizontal domains. The industrial evaluation of a hybrid system reveals that the proposed algorithm is suitable for integration into industrial systems. The algorithm is implemented as a component of Apache OpenNLP project.

Keywords Learning taxonomy • Learning syntactic parse tree • Transfer learning • Syntactic generalization • Search relevance

B.A. Galitsky (✉)
eBay Inc., San Jose, CA, USA
e-mail: boris.galitsky@ebay.com

B. Kovalerchuk
Central Washington University, Ellensburg, WA, USA
e-mail: Boris.Kovalerchuk@cwu.edu

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_21,

1 Introduction

The goal of this paper is improving relevance of web search by adding specific types of semantic information. Consider a web search task with query Q and answers $a_1, a_2, \ldots a_n$ that are produced by some conventional search engine. Assume that *an automatic semantic analysis algorithm* F is applied to the query question Q and produced a statement "Q *is about* X," e.g., "query Q is about Tax." We will denote such statement as $S(Q, X)$, where S can be viewed as a predicate "is_about(,)." Thus, $F(Q) = S(Q, X)$.

Similarly assume that the algorithm F is applied to each answer a_i and produced statements: "a_1 is about X_1," "a_2 is about X_2,"..., "a_i is about X_i,"..., "a_n is about X_n." Thus, we have $F(a_i) = S(a_i, X_i)$ for all answers. Let also L be a *score function* that measures the *similarity* between pairs $\langle Q, S(Q, X)\rangle$ and $\langle a_i, S(a_i, X_i)\rangle$, $L(\langle Q, S(Q, X)\rangle, \langle a_i, S(a_i, X_i)\rangle)$ as a mapping to the interval [0, 1]. Then we can re-rank answers a_i obtained by a conventional search engine, relative to L value. The answers with the highest scores

$$\max_{i=1,2,\ldots,n} L(\langle Q, S(Q, X)\rangle, \langle a_i, S(a_i, X_i)\rangle)$$

are considered as the *most relevant*.

The proposed measure of similarity takes into account the traditional approach, which takes all keywords from questions and answers, with our specific method of matching the keywords we determine as being essential. The above similarity will be assessed via mapping both $\langle Q, S(Q, X)\rangle$ and $\langle a_i, S(a_i, X_i)\rangle$ into a constructed structure which we refer to as *taxonomy*. Instead of keywords or bag-of-words approaches, we will also be computing similarity between parse trees for questions and answers.

This paper elaborates this approach and is organized as follows. We start from the design of the analysis algorithm F that performs the taxonomy-supported search, then we design the similarity measure L to further improve search relevance. After that we demonstrate the efficiency of the proposed method on real web searches. The paper concludes with a detailed analysis of the related works, advantages of the proposed method, and expected future work.

The algorithm uses a structured taxonomy of keywords. The taxonomy construction process starts from the seed terms (keywords) and mines the available source domains for new terms associated with these terms. These new terms are formed in several steps that involve the search engine results, parsing trees, and a machine learning algorithm. To match NL expressions between source and target domains, the algorithm uses syntactic generalization, an operation which finds a set of maximal common sub-trees of constituency parse trees of these expressions.

The industrial evaluation of a hybrid system reveals that the proposed algorithm is suitable for integration into industrial systems. The algorithm is implemented as a component of Apache OpenNLP project.

2 Method

2.1 Defining is_about Relation

The is_about relation helps to "understand" the *root concepts* of the query Q and obtain the best answer a_i. In other words, this relation sets up the set of essential keywords for Q or a_i. Let $K(Q)$ be a set of keywords of Q (the function that extracts meaningful keywords from a sentence, depending on the current choice of stop-words). $S(Q, X)$ is defined as is_about$(K(Q), X)$,

$$S(Q, X) = is\_about(K(Q), X),$$

where X is a subset of $K(Q)$. Similarly, for the answer a_i, we have $S(a_i, X_i) = is\_about(K(a_i), X)$.

Let query Q be about b and $K(Q) = \{abc\}$, i.e., $is\_about(\{a, b, c\}, b)$. Then we understand that other queries with $\{ab\}$ and $\{bc\}$ are relevant or marginally relevant to query Q, and a query with keywords $\{ac\}$ is irrelevant to Q. In other words, b is essential in Q and the other query without b term is meaningless relative to Q, and an answer which does not contain b is *irrelevant* to the query which includes b.

Example 1. Let the set of keywords {*computer, vision, technology*}, {*computer*, *vision*}, vision, technology be relevant to the query Q, and *computer*, *technology* are not, thus the query Q is about {*vision*}.

Notice that for keywords in the form of a *noun phrase* or a *verb phrase* the head or a verb may not be a keyword. Also we can group words into phrases when they form an entity, e.g., bill-gates: *is_about*({*vision, bill-gates, in-computing*}, {*bill-gates*}).

A set of keywords as called *essential* if it occurs on the right side of *is_about*. *is_about* relation as a *relation between a set of keywords and its ordered subset*.

Example 2. Let b be essential for Q with $K(Q) = \{a, b, c, d\}$, $is\_about(\{a, b, c, d\}, \{b\})$ and c also be essential when b is in the query, $is\_about(\{a, b, c, d\}, \{b, c\})$. Here b and c are ordered with b being more essential than c. Then queries and answers with $\{a, b, c\}$, $\{b, c, d\}$, $\{b, c\}$ are considered to be relevant to Q because they contain both essential keywords. In contrast, queries and answers with $\{a, b\}$, $\{b, d\}$ are considered to be (marginally) relevant to Q because c is missed and they are likely less specific. Accordingly queries and answers with $\{a, d\}$ only are considered to be irrelevant to Q.

This example gives an *idea* that we use to define the *relevance similarity score* L between the query and its answers. It is based on the order of how essential are the keywords, not only on a simple overlap of keywords or essential keywords between Q and a_i. Hence, for a query Q with $K(Q) = \{a, b, c, d\}$ and two answers (snippets) with $K(a_1) = \{b, c, d, \ldots, e, f, g\}$ and $K(a_2) = \{a, c, d, \ldots, e, f, g\}$,

the former is relevant and the latter is not because c to be essential requires such keyword b that has a higher rank of essentiality than c for the query Q with ordered essential keywords $\{b, c\}$.

Above we defined it using an *ordered set of essential keywords*. This definition works for two essential keywords. For more than two keywords we may have *multiple essentiality orders*, e.g., for three essential keywords $\{b, c, d\}$ we can have ordered sets $\{b, c\}$ and $\{b, d\}$ without $\{b, c, d\}$ that c is not required for d to be essential. To encode this essentiality order, we need to employ a tree structure, which is going to be a taxonomy. For a query, we build a structure of its keywords with essentiality relations on them by the function K, introduced above. For two keywords, one is more essential than the other if there is a path in the taxonomy tree (starting from the root) which includes the nodes with these keywords as labels, and the first node is between the second node and the root.

2.2 Defining Essential Keywords

Definition. A set of keywords E is called *essential* if it occurs on the right side of *is_about* relation and E is a *structured subset* of the set of keywords, that is the *partial* order essentiality relation "$<_E$" is defined for every pair of elements of E, $\langle E, <_E \rangle$. In this paper we limit the essentiality structure to a *tree structure* that we call a *keyword taxonomy*.

Now we need to define a method to construct essential keywords E for the query Q. Assume that all queries are from a vertical domain, e.g., tax domain. Then we can build a *common tree of concepts for the domain* (domain *taxonomy*) T, e.g., for tax domain. Later on in this paper we will describe the method for building T. The tree T includes paths which correspond with typical queries. We rely on an assumption for a vertical domain that for two words in a given domain, one word is always more essential than the other for all queries. Based on this assumption, a single taxonomy can support a search in the whole vertical domain.

Having T we can define E by set-intersection T as a set and Q, $E = T \cap Q$. This gives us the relation cover *is_about*$(K(Q), E)$. Thus one of the ways to define a query analysis algorithms F that produces *is_about* relation, from Q using T, $F(Q) =$ *is_about*$(Q, T \cap Q)$. Alternative ways to get E is to intersect keywords $K(Q)$ of Q with T, $E = T \cap K(Q)$ or to intersect Q with an individual path T_p in T, getting $E = T_p \cap Q$ and *is_about*$(Q, T_p \cap Q)$.

We say that a path T_p covers a query Q if the set of keywords for the nodes of T_p is a super-set of Q. If multiple paths cover a query Q producing different intersections $Q \cap T_p$, then this query has multiple meanings in the domain; for each such meaning a separate set of acceptable answers is expected.

The search based on such tree structures is typically referred to as the *taxonomy-based search* [7]. It identifies terms that *should* occur in the answer and terms that *must* occur there, otherwise the search result is irrelevant. The keywords that should

occur are from the taxonomy T, but the keywords that must occur are from both the query Q and taxonomy T. This is a totally different mechanism than a conventional TF*IDF-based search [21].

2.3 Constructing Relevance Score Function L

The introduction of the essentiality order for keywords leads to the specification of the relevance score function $L(\langle Q, S(Q, X)\rangle, \langle a_i, S(a_i, X_i)\rangle)$, where now X and X_i are trees of essential keywords derived by the function K. Below we discuss a method to build this function.

Consider a situation where all essential words X from the query Q are also present in the answer, $X \subseteq X_i$ the answer a_i is called *partially acceptable* for query Q. This can be defined as:

$$\text{If } X \subseteq X_i \text{ then } L(\langle Q, S(Q, X)\rangle, \langle a_i, S(a_i, X_i)\rangle) = 1.$$

However, this is a semantically shallow method especially for common situations where X and X_i contain only few words. A better way is to use a *common tree of concepts for the domain* T and use it to measure similarity between X and X_i. Now we can define L given T for the acceptable case,

$$\text{If } X \subseteq X_i \subseteq T, \text{ then } L(\langle Q, S(Q, X)\rangle, \langle a_i, S(a_i, X_i)\rangle) = 1.$$

This means that tree X is a subtree of tree X_i and both trees are subtrees of tree T.

The requirement of weak acceptability, that X is a subtree of T, is desirable, but very restrictive. Therefore, we define acceptability in the following way. An answer $a_i \in A$ is *acceptable* if it includes *all essential* (according to *is_about*) keywords from the query Q as found in the taxonomy path $T_p \subseteq T$. For any taxonomy path T_p which covers the question q (intersections of their keywords is not empty), these intersection keywords *must be* in the acceptable answer a_i.

$$\forall T_p \in T: \quad T_p \cap X \neq \emptyset \Rightarrow X_i \supseteq T_p \cap X.$$

In other words, X is a set/tree of keywords for a question, which are essential in this domain (covered by a path in the taxonomy). X also must be a subset of X_i, the set/tree of keywords for this answer i. This is a more complex requirement than $X \subseteq X_i$. $\in$ used here to denote a path in a tree. For the best answer (most accurate) we write

$$a_{\text{best}}: \quad \max_i(|X_i \cap (T_p \cap X)|), \quad T_p \in T.$$

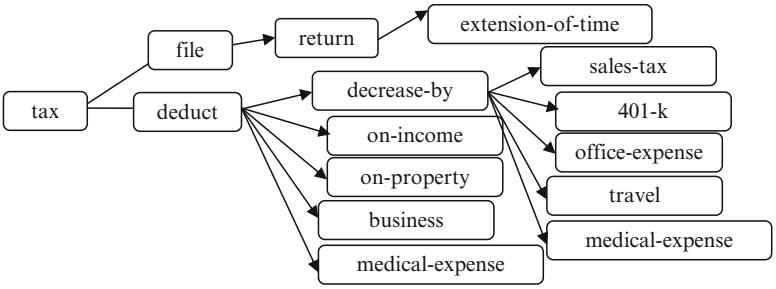


Fig. 1 An example of a snapshot of a domain taxonomy

Accordingly, we define *a taxonomy-based relevance score* L as the value of *cardinality* $|X_i \cap (T_p \cap X)|$, computed for all T_p which cover Q. Then the best answer is found among the scores for all answers A. The score L can be normalized by dividing it by $|X|$ to get it in [0, 1] interval.

The taxonomy-based score can be combined with the other scores such as TF*IDF, temporal/decay parameter, location distance, pricing, linguistic similarity, and other scores for the resultant ranking, depending on search engine architecture. In our evaluation we will be combining it with the linguistic similarity score (Sect. 2.6). Hence L indeed depends not only on X and X_i but also on Q and a_i.

Example. Consider a taxonomy T presented in Fig. 1 for the query Q = *How can tax deduction be decreased by ignoring office expense*, with a set of keywords $K(Q) = \{$*how, can, tax, deduct(ion), decreas(ed)-by, ignor(ing), office, expense*$\}$ and a set of tree answers $A = a_1, a_2, a_3$ presented as keywords:

$a_1 = \{$*deduct, tax, business, expense, while, decreas(ing), holiday, travel, away, from, office*$\}$,
$a_2 1 = \{$*pay, decreas(ed), sales-tax, return, trip, from, office, to, holiday, no, deduct(ion)*$\}$,
$a_3 1 = \{$*when, file, tax, return, deduct, decrease-by, not, calculate, office, expense, and, employee, expense*$\}$.

Notice that a_2 includes the multiword sales-tax from the taxonomy, which is also counted as a set of two words $\{$*sales, tax*$\}$. However, in a_3 decrease-by is considered as a single word because our function K considers prepositions as stop words and does not count them separately. We show ending in brackets for convenience, omitting tokenization and word form normalization. In terms of keyword overlap, a_1, a_2, and a_3 all look like good answers.

In accordance with given T query Q is covered by the path $T_p = \{\langle tax\rangle - \langle deduct\rangle - \langle decrease\text{-}by\rangle - \langle office\text{-}expense\rangle\}$.

We calculate the similarity score for each answer with Q:

$$score(a_1) = cardinality(a_1 \cap (T_p \cap Q)) = cardinality(\{tax, deduct\}) = 2;$$

$$score(a_2) = cardinality(\{tax, deduct\}) = 2;$$

$$score(a_3) = cardinality(\{tax, deduct, decrease\text{-}by, office\text{-}expense\}) = 3;$$

The answer a_3 is the best answer in this example. Our next example is about disambiguation of keywords.

Example. Consider a query q = *"When can I file extension of time for my tax return?"* with two answers:

a_1 = *"You need to file form 1234 to request a 4-month extension of time to file your tax return"*
a_2 = *"You need to download file with extension "pdf", print and complete it to do your taxes"*

and the closest taxonomy path: $Tp = \{\langle tax\rangle - \langle file\rangle - \langle return\rangle - \langle extension\text{-}of\text{-}time\rangle\}$. In this example both a_1 and a_2 contain word extension, but the keyword is "extension-of-time" not extension. Resolving this ambiguity leads to a higher score for a_1.

2.4 Taxonomy-Based Relevance Verification Algorithm

We now outline the algorithm, which takes a query Q, runs a search (outside of this algorithm), gets a set of candidate answers A, and finds the best acceptable answer according to the definitions given above.

The input: query Q
The output: the best answer a_{best} and the set of acceptable answers A_a

---

1. For a query Q, obtain a set of candidate answers A by available means (using keywords, using internal index, or using external index of search engine's APIs);
2. Find a path of taxonomy T_p which covers maximal number of terms in Q, along with other paths, which cover Q, to form a set $P = \{T_{p1}, T_{p2}, \dots\}$. Unless acceptable answer is found:
3. Compute the set $T_p \cap Q$. For each answer $a_i \in A$
4. Compute $a_i \in (T_p \in Q))$ and test if all essential words from the query, which exists in T_p are also in the answer (acceptability test)
5. Compute similarity score of Q with or each a_i
6. Compute the best answer a_{best} and the set of acceptable answers A_a. If no acceptable answer found, return to 2 for the next path from P.
7. Return a_{best} and the set of acceptable answers A_a if available

---

This algorithm filters out irrelevant answers by searching for taxonomy path (down to a leaf node if possible) which is closest to the given query in terms of the number of entities from this query. Then this path and leave node specify most accurate meaning of the query, and constrain which entities *must* occur and which *should* occur in the answer to be considered relevant. If the n-th node entity from the question occurs in answer, then all $k < n$ entities should occur in it as well.

2.5 Building Taxonomy

The domain tree/taxonomy is built iteratively. The algorithm starts from a "taxonomy seed" that contains at least two to three initial nodes (keywords) including the root of the tree. Each next iteration step k adds edges and nodes to specify existing nodes known at the step $k - 1$. A seed taxonomy can be made manually. An alternative way is using external sources, such as a glossary of that knowledge domain, e.g., http://www.investopedia.com/categories/taxes.asp for tax domain.

Example. The seed for the tax domain can contain tax as a root of the tree (a domain-determining entity), and {*deduct, income, property*} as nodes of the next level of the tree that are main entities in this domain. Another option for the seed taxonomy is presented in Fig. 1 with tax as a root, and file and deduct as child nodes at the next level.

Each iteration step is accomplished as a *learning* step using *web mining* to learn next nodes such as *sales-tax, 401k*, etc. (Fig. 1).

The learning algorithm:

(1) takes a pair (root node, child node), such as (*tax, deduct*),
(2) uses it as a search query via a search engine, e.g., Bing,
(3) extracts words and expressions which are *common* among search results (Sect. 2.6),
(4) expands the tree with these common words as new terminal nodes,
(5) takes a triple of nodes (node, child, grandchild) and repeat steps (2)–(4) for this triple.

This process can continue for k-tuples (paths on the tree) with $k > 3$ to get a deeper domain taxonomy. Here common words are single verbs, nouns, adjectives and even adverbs, prepositional phrases or multi-words, including propositional, noun, and verb phrases, which occur in *multiple* search results. The details of the extraction of common expressions between search results are explained later.

Example. Figure 2 shows some search results on Bing.com for the tree path tax-deduct-decrease. For example, for the path *tax - deduct* newly learned entities can be

$$tax - deduct \rightarrow decrease\text{-}by \qquad tax - deduct \rightarrow of\text{-}income$$

$$tax - deduct \rightarrow property\text{-}of \qquad tax - deduct \rightarrow business$$

$$tax - deduct \rightarrow medical\text{-}expense.$$

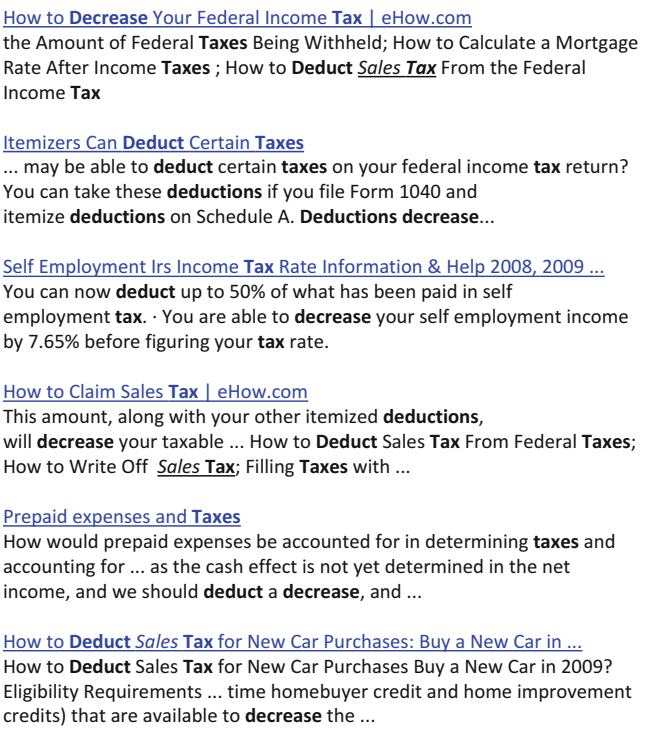


Fig. 2 Search results on Bing.com for the current taxonomy tree path tax-deduct-decrease

The format here is *existing_entity* → *new_entity*, "→" here is an unlabeled edge of the taxonomy extension at the current learning step.

Next we run triples getting:

$$\begin{aligned}
tax - deduct\text{-}decrease\text{-}by &\rightarrow sales \\
tax\text{-}deduct\text{-}decrease &\rightarrow sales\text{-}tax \\
tax\text{-}deduct\text{-}decrease\text{-}by &\rightarrow 401\text{-}K \\
tax\text{-}deduct\text{-}decrease\text{-}by &\rightarrow medical \\
tax\text{-}deduct - of\text{-}income &\rightarrow rental \\
tax\text{-}deduct - of\text{-}income &\rightarrow itemized \\
tax\text{-}deduct - of\text{-}income &\rightarrow mutual\text{-}funds
\end{aligned}$$

We outline the iterative algorithm, which takes a taxonomy with its terminal nodes and attempts to extend them via web mining to acquire a new set of terminal nodes. At the iteration k we acquire a set of nodes, extending current terminal node t_i with $t_{ik1}, t_{ik2}, \ldots$. This algorithm is based on the operation of generalization, which takes two texts as sequences $\langle$*lemma(word)*, *part-of-speech*$\rangle$ and gives least general set of texts in this form (Sect. 2.6). We outline the iterative step:

The input: Taxonomy T_k with terminal nodes $\{t_1, t_2, \ldots, t_n\}$
A threshold for the number of occurrences to provide sufficient evidence for inclusion into T_k: $th(k, T)$.
The output: extended taxonomy T_{k+1} with terminal nodes $\{t_{1k1}, t_{1k2}, \ldots, t_{2k1}, t_{2k2}, \ldots t_{nk1}, t_{nk2}\}$

For each terminal node t_i:

1. Form a search query as a path from the root to t_i, $q = \{t_{\text{root}}, \ldots, t_i\}$;
2. Run web search for q and get a set of answers (snippets) A_q.
3. Compute a pair-wise generalization (Sect. 2.6) for answers A_q: $\Lambda(A_q) = a_1 \wedge a_2, a_1 \wedge a_3, \ldots, a_1 \wedge a_m, \ldots, \ldots, a_{m-1} \wedge a_m$,
4. Sort all elements (words, phrases) of $\Lambda(A_q)$ in descending order of the number of occurrences in $\Lambda(A_q)$. Retain only the elements of $\Lambda(A_q)$ with the number of occurrences above a threshold $th(k, T)$. We call this set $\Lambda^{\text{high}}(A_q)$.
5. Subtract the labels from all existing taxonomy nodes from $\Lambda^{\text{high}}(A_q)$: $\Lambda^{\text{new}}(A_q) = \Lambda^{\text{high}}(A_q)/T_k$. We maintain the uniqueness of labels of taxonomy to simplify the online matching algorithm.
6. For each element of $\Lambda^{\text{high}}(A_q)$, create a taxonomy node t_{ihk}, where $h \in \Lambda^{\text{high}}(A_q)$, and k is the current iteration number, and add the taxonomy edge (t_i, t_{ihk}) to T_k.

The input: Taxonomy To with nodes $\{t_1, t_2, \ldots, t_n\}$ which are main entities.
The output: resultant taxonomy T with terminal nodes

Iterate through k:
Apply iterative step to k. If T_{k+1} has an empty set of nodes to add, stop

The default value of $th(k, T)$ is 2. However there is an empirical limit on how many nodes are added to a given terminal node at each iteration. This limit is 5 nodes per iteration, so we take the five highest numbers of occurrences of a term in distinct search results. This constraint helps to maintain the tree topology for the taxonomy being learned. The resultant taxonomy is a tree which is neither binary nor sorted or balanced.

Given the algorithm for the iteration step, we apply it to the set of main entities at the first step, to build the whole taxonomy:

2.6 Algorithm to Extract Common Words/Expressions Among Search Results

The word extraction algorithm is applied to a pair of sentences from two search results. These sentences are viewed as parsing trees. The algorithm produces a set of maximal common parsing sub-trees that constitute structured set of common words in these sentences. We refer the reader to further details in [8, 12].

For a given pair of words, only a single generalization exists; if words are the same in the same form, the result is a node with this word in this form. We refer to the generalization of words occurring in a syntactic tree as a *word node*. If the word forms are different (e.g., one is single and the other is plural), only the lemma of the word remains. If the words are different and only the parts of speech are the same, the resultant node contains only the part-of-speech information with no lemma. If the parts of speech are different, the generalization node is empty.

For a pair of phrases, the generalization includes all the *maximum* ordered sets of generalization nodes for the words in the phrases so that the order of words is retained. Consider the following example:

To buy the digital camera today, on Monday
The digital camera was a good buy today, the first Monday of the month

The generalization is {⟨*JJ-digital, NN-camera*⟩, ⟨*NN-today, ADV, NN -Monday*⟩}, where the generalization for the noun phrase is followed by the generalization for the adverbial phrase. The verb *buy* is excluded from both generalizations because

The input: a pair of sentences
The output: a set of maximal common sub-trees

1. Obtain the parsing tree for each sentence, using OpenNLP. For each word (tree node), we have a lemma, a part of speech and the form of the word's information. This information is contained in the node label. We also have an arc to the other node.
2. Split sentences into sub-trees that are phrases for each type: verb, noun, prepositional and others. These sub-trees are overlapping. The sub-trees are coded so that the information about their occurrence in the full tree is retained.
3. All the sub-trees are grouped by phrase types.
4. Extend the list of phrases by adding equivalence transformations [9].
5. Generalize each pair of sub-trees for both sentences for each phrase type.
6. For each pair of sub-trees, yield an alignment [14], and generalize each node for this alignment. Calculate the score for the obtained set of trees (generalization results).
7. For each pair of sub-trees of phrases, select the set of generalizations with the highest score (the least general).
8. Form the sets of generalizations for each phrase type whose elements are the sets of generalizations for that type.
9. Filter the list of generalization results: for the list of generalizations for each phrase type, exclude more general elements from the lists of generalization for a given pair of phrases.

it occurs in a different order in the above phrases. *Buy-digital-camera* is not a generalization phrase because *buy* occurs in a different sequence in the other generalization nodes.

We can see that the multiple maximum generalizations occur depending on how the correspondence between words is established; multiple generalizations are possible. Ordinarily, the total of the generalizations forms a lattice. To obey the condition of the maximum, we introduce a score for generalization. The scoring weights of generalizations are decreasing, roughly, in the following order: nouns and verbs, other parts of speech, and nodes with no lemma, only a part of speech. In its style, the generalization operation follows the notion of the "least-general generalization" or anti-unification, if a node is a formula in a language of logic. Therefore, we can refer to the syntactic tree generalization as an operation of *anti-unification of syntactic trees*.

3 Results on Improving Web Search Relevance

3.1 An Example of Taxonomy Learning Session

Let $G = \{g_i\}$ be a fixed set of linguistic relations between the pairs of terms. For instance, we may have a relation g_i(taxdeduction, reduces). In this relation "reduces" serves as an *attribute* of "tax deduction" term. An attribute of the term t that occurs in more than one answer is called a *parameter* of t.

Consider a seed taxonomy as a pair (*tax-deduct*). The taxonomy learning session consists of the following steps:

1. Getting search results $A = \{a_i\}$ for the pair (*tax-deduct*) using some web search engine.
2. Finding in each search results A_i the terms that are candidate attributes of "tax" and/or "deduct" (highlighted in Fig. 3).
3. Turning candidate attributes into parameters of "tax" and "deduct" (common attributes between different search results a_i in A) that are highlighted in dark-gray, like "overlook."
4. Extending the pair (*tax-deduct*) to a number of triples by adding, in particular, the newly acquired attribute "overlook": *Tax-deduct-overlook*.

Figure 4 shows some answers for the extended path *tax-deduct-overlook*.
The learning steps now are as follows:

1. Get search results for "tax deduct overlook";
2. Select candidate attributes (now, modifiers of terms from the current taxonomy path)
3. Turn candidate attributes into parameters by finding common expressions such as "PRP-mortgage" in our case
4. Extend the taxonomy path by adding newly acquired parameters

Fig. 3 First step of taxonomy learning, given the seed tax-deduct

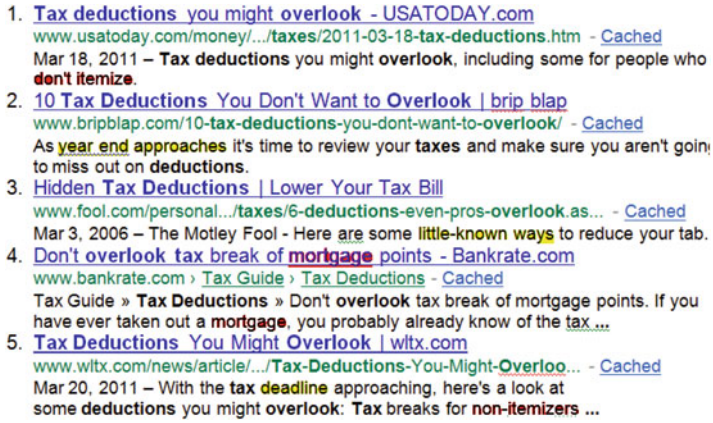


Fig. 4 Search of extension of the taxonomy tree for the path tax-deduct-overlook

Tax-deduct-overlook - ***mortgage***,
Tax-deduct-overlook - ***no_itemize***.

Having built the full taxonomy, we can now apply it to filter out search results, which are not *covered by* the taxonomy paths properly. Consider a query *Can I deduct tax on mortgage escrow account?* Fig. 5 shows the answers obtained. Two answers (shown in an oval frame) are irrelevant, because they do not include the

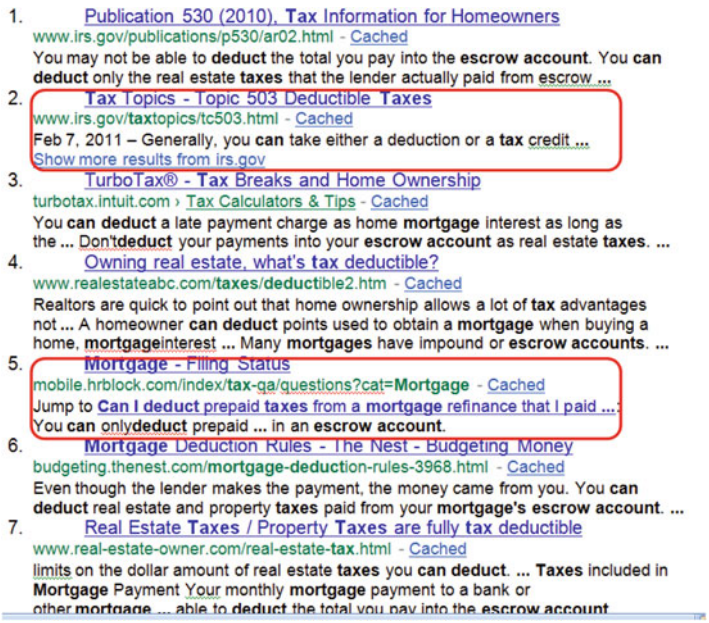


Fig. 5 Filtering out irrelevant Google answers using the built taxonomy

sell_hobby=>[[deductions, collection], [making, collection], [sales, business, collection], [collectibles, collection], [loss, hobby, collection], [item, collection], [selling, business, collection], [pay, collection], [stamp, collection], [deduction, collection], [car, collection], [sell, business, collection], [loss, collection]]

benefit=>[[office, child, parent], [credit, child, parent], [credits, child, parent], [support, child, parent], [making, child, parent], [income, child, parent], [resides, child, parent], [taxpayer, child, parent], [passed, child, parent], [claiming, child, parent], [exclusion, child, parent], [surviving, benefits, child, parent], [reporting, child, parent]]

hardship=>[[apply, undue], [taxpayer, undue], [irs, undue], [help, undue], [deductions, undue], [credits, undue], [cause, undue], [means, required, undue], [court, un-due]].

Fig. 6 Three sets of paths for the tax topic entities sell hobby, benefit, hardship

taxonomy nodes {*deduct, tax, mortgage, escrow_account*}. Notice that the closest taxonomy path to the query is *tax - deduct - overlook - mortgage - escrow_account*.

Figure 6 shows the snapshot of taxonomy tree for three entities. For each entity, given the sequence of keywords, the reader can reconstruct the meaning in the context of tax domain. This snapshot illustrates the idea of taxonomy-based search relevance improvement: once the particular meaning (content, taxonomy

path in our model) is established, we can find relevant answers. The head of the expression occurs in every path it yields (like {*sell_hobby - deductions - collection*}, {*sell_hobby - making - collection*}).

3.2 Generalization of Sentences to Extract Keywords

Consider three sentences:

I am curious how to use the digital zoom of this camera for filming insects.
How can I get short focus zoom lens for digital camera?
Can I get auto focus lens for digital camera?

Figure 7 shows the parse trees for these sentences. We generalize them by determining their maximal common sub-trees. Some sub-trees are shown as lists for brevity. The second and third trees are quite similar. Therefore, it is simple to build their common sub-tree as an (interrupted) path of the tree (Figs. 7 and 8)

{ *MD-can, PRP-I, VB-get, NN-focus, NN-lens, IN-for JJ-digital NN-camera*}.

At the phrase level, we obtain:
`Noun phrases: [ [NN-focus NN-* ], [JJ-digital NN-camera ]]`
`Verb phrases: [ [VB-get NN-focus NN-* NN-lens IN-for JJ-digital NN-camera ]]`. Here the generalization of distinct values is denoted by "*".

The common words remain in the maximum common sub-tree, except "can," which is unique to the second sentence, and the modifiers of "lens," which are different between the two sentences (shown as *NN-focus NN-* NN-lens*). We generalize sentences that are less similar than sentences two and three on a phrase-by-phrase basis. Below, the syntactic parse tree is expressed via chunking [2], using the format <posi-tion (POS - phrase)>

Parse 1 `0(S-I am curious how to use the digital zoom of this camera for filming insects), 0(NP-I), 2(VP-am curious how to use the dig-ital zoom of this camera for filming insects), ... 2(VBP-am),`

Parse 2 `[0(SBARQ-How can I get short focus zoom lens for digital camera), 0(WHADVP-How), 0(WRB-How), 4(SQ-can I get short focus zoom lens for digital camera), 4(MD-can), 8(NP-I), 8(PRP-I), 10(VP-get short focus zoom lens for digital camera),...`

Next, we group the above phrases by their phrase type [NP, VP, PP, ADJP, WHADVP]. The numbers at the beginning of each phrase encode their character positions. Each group contains the phrases of the same type because the matches occur between the same types.

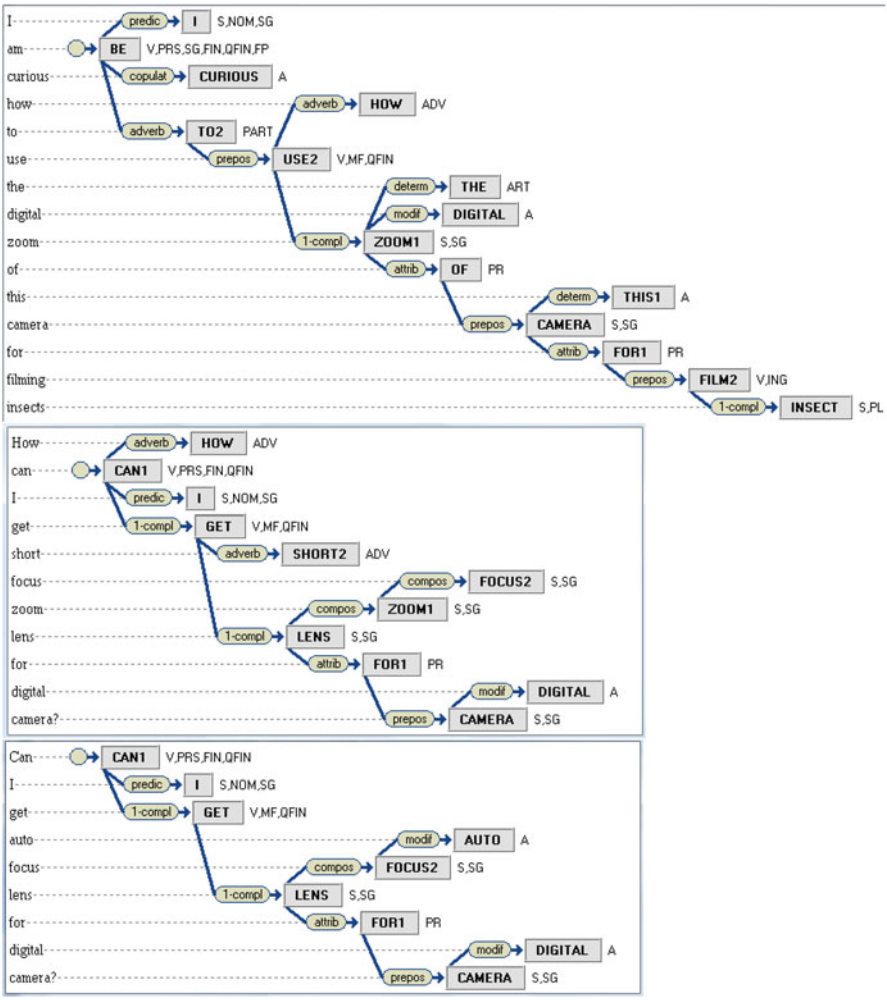


Fig. 7 Parse trees for three sentences. The *curve* shows the common sub-tree (in this case, there is only one) for the second and third sentences

Grouped Phrases 1

```
[[NP [DT-the JJ-digital NN-zoom IN-of DT-this NN-camera
], NP [DT-the JJ-digital NN-zoom ], NP [DT-this
NN-camera ], NP [VBG-filming NNS-insects ]], [VP
[VBP-am ADJP-curious WHADVP-how TO-to VB-use DT-the
JJ-digital NN-zoom IN-of DT-this NN-camera IN-for
VBG-filming NNS-insects ], VP [TO-to VB-use DT-the
JJ-digital NN-zoom IN-of DT-this NN-camera IN-for
```

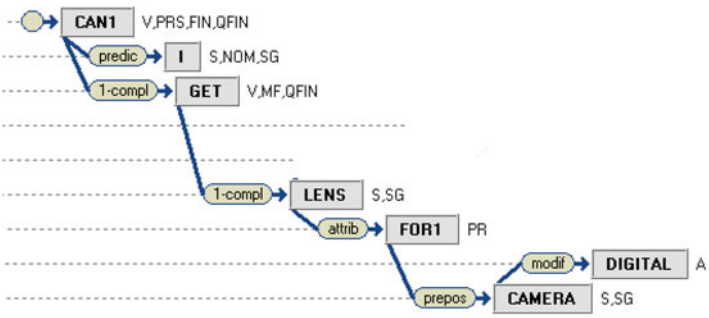


Fig. 8 Generalization results for the second and third sentences

```
VBG-filming NNS-insects ], VP [VB-use DT-the JJ-digital
NN-zoom IN-of DT-this NN-camera IN-for VBG-filming
NNS-insects]]]
```

Grouped Phrases 2

```
[[NP [JJ-short NN-focus NN-zoom NN-lens ], NP
[JJ-digital NN-camera ]], [VP [VB-get JJ-short NN-focus
NN-zoom NN-lens IN-for JJ-digital NN-camera ]], [], [PP
[IN-for JJ-digital NN-camera ]],]
```

Generalization Between Phrases

The resultant generalization is shown in bold below for verb phrases (VP).

Generalization Result

```
NP [ [JJ-* NN-zoom NN-* ], [JJ-digital NN-camera ]]
VP [ [VBP-* ADJP-* NN-zoom NN-camera ], [VB-* JJ-*
NN-zoom NN-* IN-for NN-* ]
PP [ [IN-* NN-camera ], [IN-for NN-* ]]
```

Next we compute score for the generalizations:

$\mathtt{score(NP)} = (\mathtt{W}_{<POS,*>} + \mathtt{W}_{NN} + \mathtt{W}_{<POS,*>}) + (\mathtt{W}_{NN} + \mathtt{W}_{NN}) = 3.4$,
$\mathtt{score(VP)} = (2*\mathtt{W}_{<POS,*>} + 2*\mathtt{W}_{NN}) + (4\mathtt{W}_{<POS,*>} + \mathtt{W}_{NN} + \mathtt{W}_{PRP}) = 4.55$, and
$\mathtt{score(PRP)} = (\mathtt{W}_{<POS,*>} + \mathtt{W}_{NN}) + (\mathtt{W}_{PRP} + \mathtt{W}_{NN}) = 2.55$, therefore, $\mathtt{score} = 10.5$.

Thus, such a common concept as *digital camera* is automatically generalized from the examples, as well as the verb phrase *be some-kind-of zoom camera*. The latter generalization expresses a common meaning between this pair of sentences.

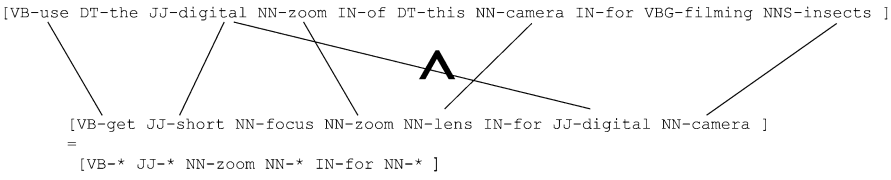
Note the occurrence of the expression [digital-camera] in the first sentence: although *digital* does not refer to *camera* directly, when we merge the two noun groups, *digital* becomes one of the adjectives of the resultant noun group with the head *camera*. It is matched against the noun phrase reformulated in a similar way (but with the preposition *for*) in the second sentence with the same head noun *camera*.

At the phrase level, generalization starts from setting correspondences between as many words as possible in the two phrases. Two phrases are aligned only if their head nouns are matched. A similar integrity constraint applies to aligning verb phrases, prepositional phrases, and other types of phrases.

We now generalize two phrases and denote the generalization operator as "∧." Six mapping links between the phrases correspond to the six members of the generalization result phrase

```
[VB-* JJ-* NN-zoom NN-* IN-for NN-* ]
```

Notice that only NN-zoom and IN-for remain as the same words, for the rest only part-of-speech information is retained.



3.3 Evaluation of Search Relevance Improvement

3.3.1 Experimental Setup

We evaluated relevance of taxonomy and syntactic generalization-enabled search engine based on Yahoo and Bing search engine APIs for the vertical domains of tax, investment, and retirement [7].

For an individual query, the *relevance* was estimated as a percentage of correct hits among the first ten hits, using the values: {correct, marginally correct, incorrect} that is in line with the approach in [27]. *Accuracy of a single search* session is calculated as the percentage of correct search results plus half of the percentage of marginally correct search results. *Accuracy of a particular search setting* (query type and search engine type) is calculated, averaging through 20 search sessions.

We also used customers' queries to eBay entertainment and product-related domains, from simple questions referring to a particular product, a particular user need, as well as a multi-sentence forum-style request to share a recommendation. The set of queries was split into noun-phrase class, verb-phrase class, how-to class, and also independently split in accordance with query length (from three keywords to multiple sentences). We ran 450 search sessions for evaluations of Sects. 2.4 and 2.6.

To compare the relevance values between search settings, we used first 100 search results obtained for a query by Yahoo and Bing APIs, and then re-sorted them according to the score of the given search setting (*syntactic generalization* score and *taxonomy-based score*). To evaluate the performance of such a hybrid system, we used the *weighted sum of these two scores* (the weights were optimized in an earlier search sessions).

3.3.2 Evaluation of Improvement of Vertical Search

Table 1 shows the results of evaluation of search relevance in the domain of vertical product search. One can see that taxonomy contributes significantly in relevance improvement, compared to domain-independent syntactic generalization. Relevance of the hybrid system that combines both of these techniques is improved by 14.8 ± 1.1 %.

The general conclusion is that for a vertical domain, a taxonomy should be definitely applied, and the syntactic generalization possibly applied, for improvement of relevance for all kinds of questions. Notice from the Table 1 results that syntactic generalization usually improves the relevance on its own, and as a part of a hybrid system in a vertical domain where the taxonomy coverage is good (most questions are mapped well into taxonomy).

Taxonomy-based method is always helpful in a vertical domain, especially for a short queries (where most keywords are represented in the taxonomy) and multi-sentence queries (where the taxonomy helps to find the important keywords for matching with a question).

3.3.3 Evaluation of Horizontal Web Search Relevance Improvement

In a horizontal domain (searching for broad topics in finance-related and product-related domains of eBay) contribution of taxonomy is comparable to syntactic generalization (Table 2). Search relevance is improved by a 4.6 ± 0.8 % by a hybrid system and is determined by a type of phrase and a query length.

The highest relevance improvement is for longer queries and for multi-sentence queries. Noun phrases perform better at the baseline (Yahoo and Bing search engine

Table 1 Improvement of accuracy in a vertical domain

Query	Phrase sub-type	Relevancy of baseline Yahoo search, %, averaging over 20 searches	Relevancy of baseline Bing search, %, averaging over 20 searches	Relevancy of re-sorting by generalization, %, averaging over 20 searches	Relevancy of re-sorting by using taxonomy, %, averaging over 20 searches	Relevancy of re-sorting by using taxonomy and generalization, %, averaging over 20 searches	Relevancy improvement for hybrid approach, comp. to baseline (averaged for Bing & Yahoo)
3–4 word phrases	Noun phrase	86.7	85.4	87.1	93.5	93.6	1.088
	Verb phrase	83.4	82.9	79.9	92.1	92.8	1.116
	How-to expression	76.7	78.2	79.5	93.4	93.3	1.205
	Average	82.3	82.2	82.2	93.0	93.2	1.134
5–10 word phrases	Noun phrase	84.1	84.9	87.3	91.7	92.1	1.090
	Verb phrase	83.5	82.7	86.1	92.4	93.4	1.124
	How-to expression	82.0	82.9	82.1	88.9	91.6	1.111
	Average	83.2	83.5	85.2	91.0	92.4	1.108
2–3 sentences	One verb one noun phrases	68.8	67.6	69.1	81.2	83.1	1.218
	Both verb phrases	66.3	67.1	71.2	77.4	78.3	1.174
	One sent of how-to type	66.1	68.3	73.2	79.2	80.9	1.204
	Average	67.1	67.7	71.2	79.3	80.8	1.199

The 3rd and 4th columns on the left are the baseline; the right-most column shows improvement of search accuracy

Table 2 Evaluation of search relevance improvement in a horizontal domain

Query	Phrase sub-type	Relevancy of baseline Yahoo search, %, averaging over 20 searches	Relevancy of baseline Bing search, %, averaging over 20 searches	Relevancy of re-sorting by generalization, %, averaging over 20 searches	Relevancy of re-sorting by using taxonomy, %, averaging over 20 searches	Relevancy of re-sorting by using taxonomy and generalization, %, averaging over 20 searches	Relevancy improvement for hybrid approach, comp. to baseline (averaged for Bing & Yahoo)
3–4 word phrases	Noun phrase	88.1	88.0	87.5	89.2	89.4	1.015
	Verb phrase	83.4	82.9	79.9	80.5	84.2	1.013
	How-to expression	76.7	81.2	79.5	77.0	80.4	1.018
	Average	82.7	84.0	82.3	82.2	84.7	1.015
5–6 word phrases	Noun phrase	86.3	85.4	87.3	85.8	88.4	1.030
	Verb phrase	84.4	85.2	86.1	88.3	88.7	1.046
	How-to expression	83.0	82.9	82.1	84.2	85.6	1.032
	Average	84.6	84.5	85.2	86.1	87.6	1.036
7–8 word phrases	Noun phrase	78.4	79.3	81.1	82.8	83.0	1.053
	Verb phrase	75.2	73.8	74.3	78.3	79.2	1.063
	How-to expression	73.2	73.9	74.5	77.8	76.3	1.037
	Average	75.6	75.7	76.6	79.6	79.5	1.051
8–10 word single sentences	Noun phrase	68.8	67.9	71.2	69.7	72.4	1.059

(continued)

Table 2 (continued)

Query	Phrase sub-type	Relevancy of baseline Yahoo search, %, averaging over 20 searches	Relevancy of baseline Bing search, %, averaging over 20 searches	Relevancy of re-sorting by generalization, %, averaging over 20 searches	Relevancy of re-sorting by using taxonomy, %, averaging over 20 searches	Relevancy of re-sorting by using taxonomy and generalization, %, averaging over 20 searches	Relevancy improvement for hybrid approach, comp. to baseline (averaged for Bing & Yahoo)
	Verb phrase	65.8	67.2	73.6	70.2	73.1	1.099
	How-to expression	64.3	63.9	65.7	67.5	68.1	1.062
	Average	66.3	66.3	70.2	69.1	71.2	1.074
2 Sentences, > 8 words total	1 verb 1 noun phrases	66.5	67.2	66.9	69.2	70.2	1.050
	Both verb phrases	65.4	63.9	65.0	67.3	69.4	1.073
	One sent of how-to type	65.9	66.7	66.3	65.2	67.9	1.024
	Average	65.9	65.9	66.1	67.2	69.2	1.049
3 sentences, > 12 words total	1 verb 1 noun phrases	63.6	62.9	64.5	65.2	68.1	1.077
	Both verb phrases	63.1	64.7	63.4	62.5	67.2	1.052
	One sent of how-to type	64.2	65.3	65.7	64.7	66.8	1.032
	Average	63.6	64.3	64.5	64.1	67.4	1.053

APIs) and a hybrid system, than the verb phrases and the how-to phrases. Note that generalization can decrease relevance for short queries, where linguistic information is not as important as frequency analysis.

One can see from Table 2 that the hybrid system almost always outperforms the individual components. Thus, for a horizontal domain, syntactic generalization is a must and taxonomy is helpful for some queries, which happen to be covered by this taxonomy, and is useless for the majority of queries.

We observed that a taxonomy is beneficial for queries in many forms, and their complexities. In contrast, syntactic generalization-supported search is beneficial for rather complex queries, exceeding three to four keywords. Taxonomy-based search is essential for a product search without requiring explicit use of their features and/or needs. Conversely, syntactic generalization is sensitive to proper handling of phrasings in product names, matching the template:

product_name for *super-product* with *parts-of-product*.

We conclude that building taxonomy for such domain as product search is a plausible and rewarding task, and should be done for all kinds of product searches.

3.4 Multi-Lingual Taxonomy Use

Syntactic generalization was deployed and evaluated in the framework of a Unique European Citizens' attention service (iSAC6+) project, an EU initiative to build a recommendation search engine in a vertical domain. As a part of this initiative, a taxonomy was built to improve the search relevance (easy4.udg.edu/isac/eng/index.php, [5]). Taxonomy learning of the tax domain was conducted in English and then translated to Spanish, French, German, and Italian. It was evaluated by project partners using the tool in Figs. 9 and 10. To improve search precision a project partner in a particular location modifies the automatically learned taxonomy to fix a particular case, upload the taxonomy version adjusted for a particular location (Fig. 9) and verify the improvement of relevance. An evaluator is able to sort search results by the original Yahoo score, the syntactic generalization score, and the taxonomy score to get a sense of how each of these scores works and how they correlate with the best order of answers for the best relevance (Fig. 10).

3.5 Commercial Evaluation of Taxonomy-Based Text Similarity

We subject the proposed technique of taxonomy-based and syntactic generalization-based techniques to commercial mainstream news analysis at AllVoices.com

Fig. 9 Tool for manual taxonomy adjustment for citizens recommendation services

Can Form 1040 EZ be used to claim the earned income credit

You can change the ordering of the table by clicking on column-headers.

First result Previous result Next result Last result

ORIGINAL-RANK	SYNTACTIC-MATCH SCORE	TAXONOMY-SCORE	TITLE & ABSTRACT
16	3.3	1	2010 **Form** W-5 Use **Form** W-5 if you are eligible to get part of th
3	3.3	4	**Earned Income Credit** **Can Form** 1040EZ **be used to claim the earned i**
2	3.3	4	**Can Form** 1040EZ **be used to claim the earned i** **Can Form** 1040EZ **be used to claim the earned i**
0	3.3	4	Other EITC Issues Question: **Can Form** 1040EZ **be used to claim th**
20	3.0	0	Line by Line Tips for **Form 1040-EZ** (Year 2008) Prepare your 2008 tax returns on **Form 1040-EZ**
5	3.0	0	Line by Line Tips for **Form 1040-EZ** (Year 2009) Prepare your 2009 tax returns on **Form 1040-EZ**
17	2.9	1	FREE 1040EZ - FREE Federal 1040EZ - Federal 10 Now, as an individual, you may wonder whether y
27	2.8	0	2007 **Form** W-5 I expect to have a qualifying child and be able to
19	2.8	1	2008 **Form** W-5 I expect to have a qualifying child and be able to

Fig. 10 Sorting search results by taxonomy-based and syntactic generalization scores for a given query "Can Form 1040 EZ be used to claim the earned income credit?"

(Fig. 11). The task is to *cluster* relevant news items together by means of finding similarity between the titles and first paragraphs, similarly to what we have done with questions and answers. By definition, multiple news articles belong to the same cluster if there is a substantial overlap in geographical locations, the names of individuals, organizations, other agents, and the relationships between them. Some of these can be extracted using entity taggers and/or taxonomies built offline, and some are handled in real time using syntactic generalization (the bottom of Fig. 12). The latter is applicable if there is a lack of prior entity information.

In addition, syntactic generalization and taxonomy match was used to aggregate relevant images and videos from different sources, such as Google Image, YouTube, and Flickr. It was implemented by assessing their relevance given their textual descriptions and tags. The precision of the text analysis is achieved by the site's usability (click rate): more than nine million unique visitors per month. If the precision were low, we assume that users would not click through to irrelevant "similar" articles. Recall is accessed manually; however, the system needs to find at

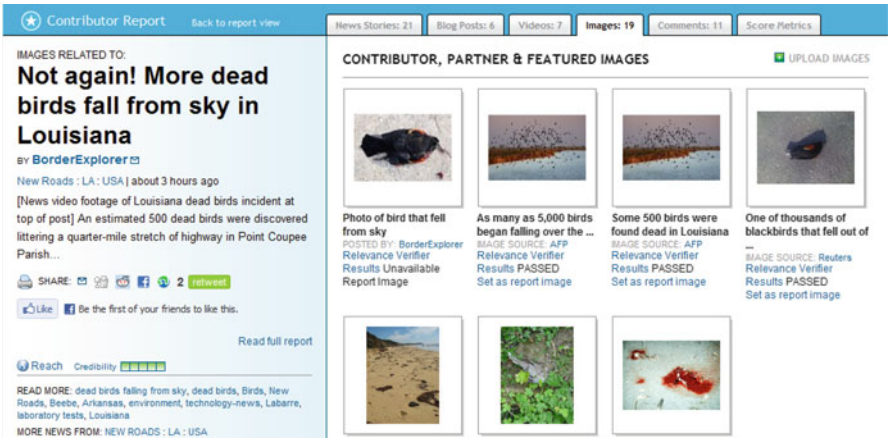


Fig. 11 News articles and aggregated images found on the web and determined to be relevant to this article

least a few articles, images, and videos for each incoming article. Recall is generally not an issue for web mining and web document analysis (it is assumed that there is a sufficiently high number of articles, images, and videos on the web for mining).

Relevance is ensured in two steps. First, we form a query to the image/video/blog search engine API, given an event title and first paragraph and extracting and filtering noun phrases by certain significance criteria. Second, we apply a similarity assessment to the texts returned from images/videos/blogs and ensure that substantial common noun, verb, or prepositional sub-phrases can be identified between the seed events and these media (Fig. 13).

The precision data for the relevance relationships between an article and other articles, blog postings, images, and videos are presented in Table 3. Note that by itself, the taxonomy-based method has a very low precision and does not outperform the baseline of the statistical assessment. This baseline is based on TF*IDF model for keyword-based assessment of relevance. However, there is a noticeable improvement in the precision of the hybrid system, where the major contribution of syntactic generalization is improved by a few percentage points by the taxonomy-based method [10, 11]. We can conclude that syntactic generalization and the taxonomy-based methods (which also rely on syntactic generalization) use different sources of relevance information. Therefore, they are complementary to each other.

The objective of syntactic generalization is to filter out false-positive relevance decisions made by a statistical relevance engines. This statistical engine has been designed following [19, 20]. The percentage of false-positive news stories was reduced from 29 to 17 % (approximately 30,000 stories/month, viewed by nine million unique users), and the percentage of false-positive image attachment was reduced from 24 to 20 % (approximately 3,000 images and 500 videos attached to stories monthly). The percentages shown are (100 %—precision values); recall

Fig. 12 Syntactic generalization result for the seed articles and the other article mined for on the web

values are not as important for web mining, assuming there is an unlimited number of resources on the web and that we must identify the relevant ones.

Our approach belongs to the category of structural machine learning. The accuracy of our approach is worth comparing with the other parse tree learning approach based on the statistical learning of SVM. For instance, Moschitti [22] compares the performances of the bag-of-words kernel, syntactic parse trees and predicate argument structures kernel, and the semantic role kernel, confirming that the accuracy improves in this order and reaches an F-measure of 68 % on the TREC dataset. Achieving comparable accuracies, the kernel-based approach requires manual adjustment. However, it does not provide similarity data in the explicit form of common sub-phrases. Structural machine learning methods are better suited for performance-critical production environments serving hundreds millions of users

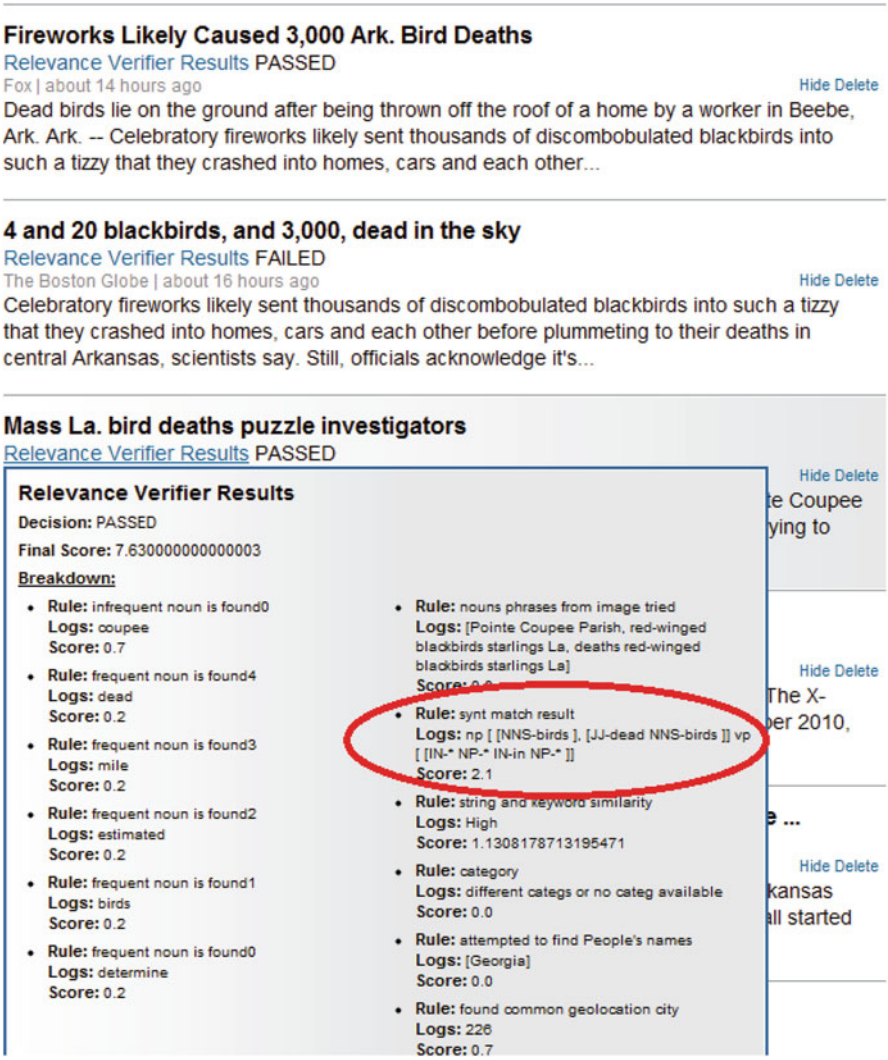


Fig. 13 Explanation for relevance decision while forming a cluster of news articles for the one in Fig. 11. The *circled area* shows the syntactic generalization result for the seed articles and the given one

because they better fit modern software quality assurance methodologies. Logs of the discovered commonality expressions are maintained and tracked, which ensures the required performance as the system evolves over time and the text classification domains change.

Table 3 Improvement in the precision of text similarity

Media/method of text similarity assessment	Full size news articles	Abstracts of articles	Blog posting	Comments	Images	Videos
Frequencies of terms in documents (baseline) (%)	29.3	26.1	31.4	32.0	24.1	25.2
Syntactic generalization (%)	19.7	18.4	20.8	27.1	20.1	19.0
Taxonomy based (%)	45.0	41.7	44.9	52.3	44.8	43.1
Hybrid syntactic generalization and taxonomy based (%)	17.2	16.6	17.5	24.1	20.2	18.0

3.6 Opinion-Oriented Open Search Engine

The search engine based on syntactic generalization is designed to provide opinion data in an aggregated form obtained from various sources. This search engine uses conventional search results and Google-sponsored link formats that are already accepted by a vast community of users.

The user interface is shown in Fig. 14. To search for an opinion, a user specifies a product class, a name of particular products, and a set of its features, specific concerns, needs, or interests. A search can be narrowed down to a particular source; otherwise, multiple sources of opinion (review portals, vendor-owned reviews, forums and blogs available for indexing) are combined.

The opinion search results are shown on the bottom left. For each result, a snapshot is generated indicating a product, its features that the system attempts to match to a user opinion request, and sentiments. In case of multiple sentence queries, a hit contains a combined snapshot of multiple opinions from multiple sources, dynamically linked to match the user request.

Automatically generated product ads compliant with the Google-sponsored link format are shown on the right. The phrases in the generated ads are extracted from the original products' web pages and may be modified for compatibility, compactness, and their appeal to potential users. There is a one-to-one correspondence between the products in the opinion hits on the left and the generated ads on the right (unlike in Google, where the sponsored links list different websites from those presented on the left).

Both respective business representatives and product users are encouraged to edit and add ads, expressing product feature highlights and usability opinions,

respectively. This feature assures openness and community participation in providing access to linked opinions for other users. A search phrase may combine multiple sentences: for example: "*I am a beginning user of digital cameras. I want to take pictures of my kids and pets. Sometimes I take it outdoors, so it should be waterproof to resist the rain*."

Obviously, this type of specific opinion request can hardly be represented by keywords like "beginner digital camera kids pets waterproof rain."

For a multi-sentence query, the results are provided as linked search hits:

> **Take Pictures** of Your **Kids**? ... Canon 400D EOS Rebel XTI **digital** SLR **camera** review ↔ I am by no means a professional or long-time user of SLR cameras.
>
> How To **Take Pictures** Of **Pets** And **Kids** ... Need help with **Digital** slr **camera** please!!!? - Yahoo! Answers ↔ I am a **beginner** in the world of the **digital** SLR ...
>
> Canon 400D EOS Rebel XTI **digital** SLR **camera** review (Website Design Tips) / Animal, **pet**, **children**, equine, livestock, farm portrait and stock ↔ I am a **beginner** to the slr **camera** world. ↔ I want to **take** the best **picture** possible because I know you.

Linking (↔) is determined in real time to address each part of a multi-sentence query, which may be a blog posting seeking advice. Linked search results provide comprehensive opinions on the topic of the user's interest, obtained from various sources and linked on the fly.

The problem of matching user needs while product search has been addressed in [3, 13]. An example of such user need expression would be "a cell phone which fits in my palm."

This need depends on the item's nature, the user's preferences, and time. Blanco-Fernández et al. [3] present a filtering strategy that exploits the semantics formalized in an ontology in order to link items (and their features) to time functions; the shapes of these functions are corrected by temporal curves built from the consumption stereotypes which are personalized to users.

4 Related Work

4.1 Transfer Learning Paradigm for Vertical and Horizontal Domains

In this paper we approached taxonomy building from the standpoint *of transfer learning paradigm* [25, 28]. Although we build our taxonomy to function in a vertical domain, we use a horizontal domain for web mining to build it. For building taxonomies, transfer learning allows knowledge to be extracted from a wide

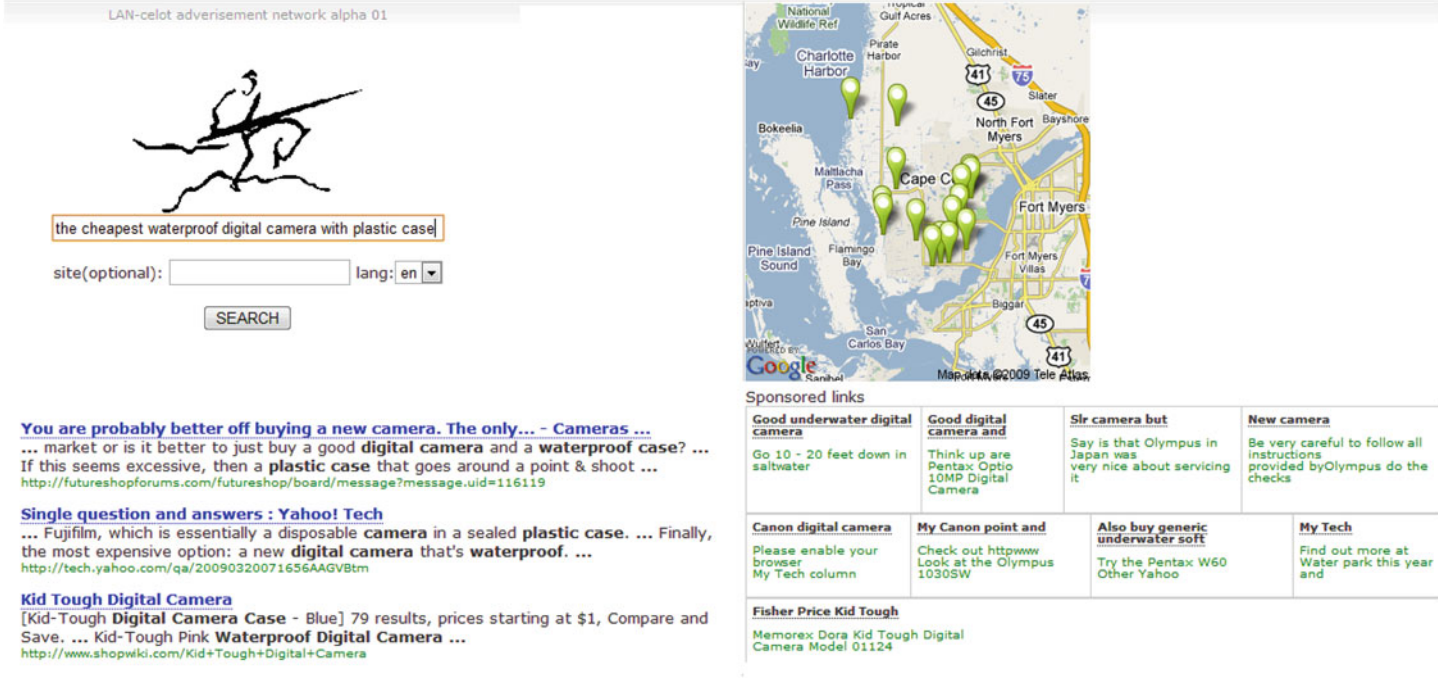


Fig. 14 User interface of the search engine used to evaluate taxonomies

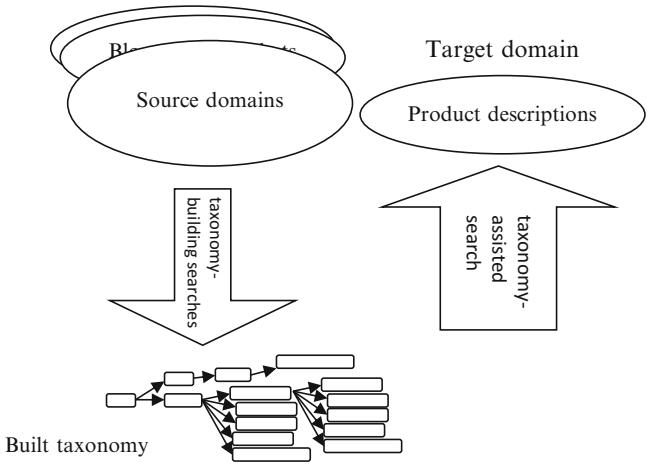


Fig. 15 Taxonomy-assisted search viewed from the standpoint of transfer learning

spectrum of web domains and be used to enhance taxonomy-based search in a target domain. We will call these web domains auxiliary domains.

For transfer learning we compute the similarity between phrases in auxiliary and target domains using syntactic generalization as an extension of bag-of-words approach. The paper introduced a novel way for finding the structural similarity between sentences, to enable transfer learning at a structured knowledge level. This allows learning a nontrivial structural (semantic) similarity mapping between phrases in two different domains when they are completely different in terms of their vocabularies.

It is usually insufficient to web mine documents for building taxonomies for vertical domains in this vertical domain only. Moreover, when a target domain includes social network data, micro-text, it is usually hard to find enough such data for building taxonomies within this domain, so a transfer learning methodology is required, which mines a wider set of domains with similar vocabulary. The transfer learning is then needed to be supported by matching syntactic expressions from distinct domains. In this study we perform it on the level of constituency parse trees (Fig. 15).

4.2 Taxonomies

WordNet is the most commonly used computational lexicon of English for word sense disambiguation, a task aimed to assigning the most appropriate senses (i.e., synsets) to words in context [23]. It has been argued that WordNet encodes sense distinctions that are too fine-grained even for humans. This issue prevents WSD systems from achieving high performance. The granularity issue has been tackled

by proposing clustering methods that automatically group together similar senses of the same word. Though WordNet contains a sufficiently wide range of common words, it does not cover special domain vocabulary. Since it is primarily designed to act as an underlying database for different applications, those applications cannot be used in specific domains that are not covered by WordNet.

A number of currently available general-purpose resources such as DBPEdia, Free-base, Yago assist entity-related searches, but are insufficient to filter out irrelevant answers concerning certain activity with an entity and its multiple parameters. A set of vertical ontologies, such as last.fm for artists, are also helpful for entity-based searches in vertical domains; however, their taxonomy trees are rather shallow, and usability for recognizing irrelevant answers is limited.

As text documents are massively available on the web as well as an access to them via **web search engine APIs**, most researchers have attempted to learn taxonomies on the basis of textual input. Several researchers explored taxonomic relations explicitly expressed in texts by pattern matching [16, 24]. One drawback of pattern matching is that it involves the predefined choice of semantic relations to be extracted.

In this study, to improve the flexibility of pattern matching we used transfer learning based on parse patterns, which is higher level of abstraction than sequences of words. We extend the notion of syntactic contexts from a partial cases such as noun + modifier and dependency triple [18] towards finding a parse sub-tree in a parse tree. Our approach also extends handling of internal structure of noun phrases used to find taxonomic relations [4]. Many researchers follow Harris' distributional hypothesis of correlation between semantic similarity of words or terms, and the extent to which they share similar syntactic contexts [15]. Clustering only requires a minimal amount of manual semantic annotation by a knowledge engineer, so clustering is frequently combined with pattern matching to be applied to syntactic contexts in order to also extract previously unexpected relations. We improve learning taxonomy on the web by combining supervised learning of the seed with unsupervised learning of the consecutive sets of relationships, also addressing such requirements of a taxonomy building process as evolvability and adaptability to new query domains of search engine users.

The current challenge in the area of taxonomy-supported searches is how to apply an imperfect taxonomy, automatically compiled from the web, to improve search. Lightweight keyword-based approaches cannot address this challenge. This paper addresses it by using web mining to get training data for learning, and syntactic generalization as a learning tool.

This paper presented an *automated taxonomy building mechanism* which is based on initial set of main entities (a seed) for given vertical knowledge domain. This seed is then automatically extended by mining of web documents which include a meaning of a current taxonomy node. This node is further extended by entities which are the results of inductive learning of commonalities between these documents. These commonalities are extracted using an operation of syntactic generalization, which finds the common parts of syntactic parse trees of a set of documents, obtained for the current taxonomy node. Syntactic generalization has

been extensively evaluated commercially to improve text relevance [9–11], and in this study we also apply it in the transfer learning setting for automated building of taxonomies.

Proceeding from *parsing to semantic level* is an important task towards natural language understanding and has immediate applications in tasks such as information extraction and question answering [1, 6, 26, 30]. In the last 10 years there has been a dramatic shift in computational linguistics from manually constructing grammars and knowledge bases to partially or totally automating this process by using statistical learning methods trained on large annotated or non-annotated natural language corpora. However, instead of using such corpora, in this paper we use web search results for common queries, since their accuracy is higher and they are more up-to-date than academic linguistic resources in terms of specific domain knowledge, such as tax.

The value of semantically enabling search engines for improving search relevance has been well understood by the commercial search engine community [17]. Once an "ideal" taxonomy is available, properly covering all important entities in a vertical domain, it can be directly applied to filtering out irrelevant answers.

4.3 Comparative Analysis of Taxonomy-Based Systems

Table 4 presents the comparative analysis of some of taxonomy-based systems. It includes description of the taxonomy type, building mode, and its properties with respect to support of search, including filtering irrelevant answers by matching them with the query and taxonomy. In this table the current approach is the only one directly targeting filtering out irrelevant answers obtained by other components of search engines.

5 Conclusion

We conclude that full-scale syntactic processing approach based on keyword taxonomy learning, and iterative taxonomy extension, is a viable way to enhance web search engines. Java-based OpenNLP component serves as an illustration of the proposed algorithm, and it is ready to be integrated with existing search engines.

In the future studies we plan to proceed from generalization of individual sentences to the level of paragraphs, deploying discourse theories and deeper analyzing the structure of text.

Table 4 Comparative analysis of taxonomy-based systems with respect to support of search

Textual inference based support of search [29]	Probabilistic approximate textual inference over tuples extracted from text. Utilizes sizable chunks of the Web corpus as source text. Taxonomy is constructed as a Markov network. The input a conjunctive query, a set of inference rules expressed as Horn clauses, and large sets of ground assertions extracted from the Web, WordNet, and other knowledge bases	No real time syntactic match is conducted	Utilizes logical inference to find the subset of ground assertions and inference rules that may influence the answers to the query—enabling the construction of a focused Markov network	Finds correct answers on its own, but does not filter incorrect ones
This work	Search-oriented taxonomies are built via web mining by employing machine learning of parse trees. Semisupervised learning setting in a vertical domains, using search engine APIs	Facilitates match between query and candidate answer	Features in parse tree and limited reasoning	Specifically designed for this purpose

References

1. Allen, J.F.: Natural Language Understanding. Benjamin Cummings, Menlo Park (1987)
2. Abney, S.: "Parsing by Chunks", Principle-Based Parsing, Kluwer Academic Publishers, pp. 257–278 (1991)
3. Blanco-Fernández, Y., López-Nores, M., Pazos-Arias, J.J., Garc'ıa-Duque, J.: An improvement for semantics-based recommender systems grounded on attaching temporal information to ontologies and user profiles. Eng. Appl. Artif. Intell. **24**(8), 1385–1397 (2011)
4. Buitelaar, P., Olejnik, D., Sintek, M.: A proteg'e' plug-in for ontology extraction from text based on linguistic analysis. In: Proceedings of the International Semantic Web Conference (ISWC) (2003)
5. De la Rosa, J.L., Rovira, M., Beer, M., Montaner, M., Gibovic, D.: Reducing administrative burden by online information and referral services. In: Reddick, C.G. (ed.) Citizens and E-Government: Evaluating Policy and Management, pp. 131–157. IGI Global, Austin (2010)
6. Dzikovska, M., Swift, M., Allen, J., de Beaumont, W.: Generic parsing for multi-domain semantic interpretation. In: International Workshop on Parsing Technologies (Iwpt05), Vancouver (2005)

7. Galitsky, B.: Natural Language Question Answering System: Technique of Semantic Headers. Advanced Knowledge International, Adelaide (2003)
8. Galitsky, B.: Machine learning of syntactic parse trees for search and classification of text. Eng. Appl. Artif. Intell. **26**(3), 1072–1091 (2013)
9. Galitsky, B., Dobrocsi, G., de la Rosa, J.L., Kuznetsov, S.O.: From generalization of syntactic parse trees to conceptual graphs. In: 18th International Conference on Conceptual Structures (ICCS), pp. 185–190 (2010)
10. Galitsky, B.A., Kovalerchuk, B., de la Rosa, J.L.: Assessing plausibility of explanation and meta-explanation in inter-human conflicts. A special issue on semantic-based information and engineering systems. Eng. Appl. Artif. Intell. **24**(8), 1472–1486 (2011)
11. Galitsky, B., Dobrocsi, G., de la Rosa, J.L., Kuznetsov, S.O.: Using Generalization of syntactic parse trees for taxonomy capture on the web. In: 19th International Conference on Conceptual Structures (ICCS), pp. 104–117 (2011)
12. Galitsky, B., Dobrocsi, G., de la Rosa, J.L.: Inferring semantic properties of sentences mining syntactic parse trees. Data Knowl. Eng. **81–82**, 21–45 (2012)
13. Galitsky, B., González, M.P, Chesñevar C.I.: A novel approach for classifying customer complaints through graphs similarities in argumentative dialogue. Decision Support Systems **46**(3), 717–729 (2009)
14. Gildea, D.: Loosely tree-based alignment for machine translation. In Proceedings of the 41th Annual Conference of the Association for Computational Linguistics (ACL-03), pp. 80–87, Sapporo, Japan (2003)
15. Harris, Z.: Mathematical Structures of Language. Wiley, London (1968)
16. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics, pp. 539–545 (1992)
17. Heddon, H.: Better living through taxonomies. Digital Web Magazine, www.digital-web.com/articles/better_living_through_taxonomies/ (2008)
18. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of COLING-ACL98 (1998)
19. Liu, J., Birnbaum, L.: Measuring semantic similarity between named entities by searching the web directory. In: Web Intelligence, pp. 461–465 (2007)
20. Liu, J., Birnbaum, L.: What do they think?: Aggregating local views about news events and topics. In: In Proceedings of the 17th International Conference on World Wide Web (WWW 2008), pp. 1021–1022 (2008)
21. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
22. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Proceedings of the 17th European Conference on Machine Learning, Berlin (2006)
23. Navigli, R.: Word sense disambiguation: a survey. ACM Comput. Surv. **41**(2), pp. 1–69 (2009)
24. Poesio, M., Ishikawa, T., Schulte im Walde, S., Viera, R.: Acquiring lexical knowledge for anaphora resolution. In: Proceedings of the 3rd Conference on Language Resources and Evaluation (LREC) (2002)
25. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: Proceedings of 24th International Conference on Machine Learning, pp. 759–766, June 2007
26. Ravichandran, D., Hovy, E.: Learning surface text patterns for a question answering system. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia (2002)
27. Resnik, P., Lin, J.: Evaluation of NLP systems. In: Clark, A., Fox, C., Lappin, S. (eds.) The Handbook of Computational Linguistics and Natural Language Processing. Wiley-Blackwell, Oxford (2010)

28. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
29. Schoenmackers, S., Etzioni, O., Weld, D.S.: Scaling textual inference to the web. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2008)
30. Wang, K., Ming, Z., Chua, T.-S.: A syntactic tree matching approach to finding similar questions in community-based QA services. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09), pp. 187–194. ACM, New York (2009)

Linear Regression via Elastic Net: Non-enumerative Leave-One-Out Verification of Feature Selection

Elena Chernousova, Nikolay Razin, Olga Krasotkina, Vadim Mottl, and David Windridge

Abstract The feature-selective non-quadratic Elastic Net criterion of regression estimation is completely determined by two numerical regularization parameters which penalize, respectively, the squared and absolute values of the regression coefficients under estimation. It is an inherent property of the minimum of the Elastic Net that the values of regularization parameters completely determine a partition of the variable set into three subsets of negative, positive, and strictly zero values, so that the former two subsets and the latter subset are, respectively, associated with "informative" and "redundant" features. We propose in this paper to treat this partition as a secondary structural parameter to be verified by leave-one-out cross validation. Once the partitioning is fixed, we show that there exists a non-enumerative method for computing the leave-one-out error rate, thus enabling an evaluation of model generality in order to tune the structural parameters without the necessity of multiple training repetitions.

Keywords Elastic Net regression • Partitioning of the feature set • Secondary structural parameter • Feature selection • Non-enumerative leave-one-out

E. Chernousova (✉) • N. Razin
Moscow Institute of Physics and Technology, Moscow, Russia
e-mail: elchernousova@inbox.ru; nrmanutd@gmail.com

O. Krasotkina
Tula State University, Tula, Russia
e-mail: o.v.krasotkina@yandex.ru

V. Mottl
Computing Centre of the Russian Academy of Sciences, Moscow, Russia
e-mail: vmottl@yandex.ru

D. Windridge
University of Surrey, Guildford, UK
e-mail: d.windridge@surrey.ac.uk

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_22,

1 Introduction

The Elastic Net regularization principle, proposed by Zou and Hastie in [1] as a generalization of Tibshirani's previous Lasso principle [2], is a convenient and effective means of feature selection in machine learning that proceeds via double penalization of both the squared and absolute values of the coefficients under estimation. It improves on alternative methods by virtue of its ability to assign strictly zero values to redundant coefficients, thereby enabling the subset of informative features to be determined without discrete search. Having been developed originally for use in regression, it was later successfully incorporated into training criteria for pattern recognition in the SVM formulation [3, 4], as well as in terms of logistic regression [5, 6].

In this paper, we restrict our attention to the problem of regression estimation. Our aim is to find a computationally effective algorithm for computing the leave-one-out error rate so as to determine model generality for tuning structural parameters while avoiding multiple training repetitions.

Assuming a centered and normalized training set

$$\{(\mathbf{x}_j, y_j), j = 1, \ldots, N\}, \mathbf{x}_j = (x_{1j} \cdots x_{nj})^T \in \mathbb{R}^n, \; y_j \in \mathbb{R}, \tag{1}$$

$$\sum_{j=1}^{N} \mathbf{x}_j = \mathbf{0}, \sum_{j=1}^{N} y_j = 0, \; \frac{1}{N} \sum_{j=1}^{N} x_{ij}^2 = 1, \; i \in I = \{1, \ldots, n\}, \tag{2}$$

the initial Elastic Net criterion, referred to in [1] as the "naive" Elastic Net, consists in estimating the real-valued coefficients $\mathbf{a} = (a_1 \cdots a_n)^T \in \mathbb{R}^n$ of the regression model $\hat{y}(\mathbf{x}) = \mathbf{a}^T \mathbf{x}$ as the minimum point of the convex objective function:

$$\begin{aligned} J_{\text{NEN}}(\mathbf{a}|\lambda_1, \lambda_2) &= \lambda_2 \sum_{i=1}^{n} a_i^2 + \lambda_1 \sum_{i=1}^{n} |a_i| + \sum_{j=1}^{N} \Big(y_j - \sum_{i=1}^{n} a_i x_{ij} \Big)^2 \\ &= \lambda_2 \mathbf{a}^T \mathbf{a} + \lambda_1 \|\mathbf{a}\|_1 + \big(\mathbf{y} - \mathbf{X}\mathbf{a}\big)^T \big(\mathbf{y} - \mathbf{X}\mathbf{a}\big) \to \min(\mathbf{a}), \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{y} &= (y_1 \cdots y_N) \in \mathbb{R}^N, \; \mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N)^T (N \times n), \\ \hat{\mathbf{a}}_{\lambda_1, \lambda_2} &= \big(\hat{a}_{i, \lambda_1, \lambda_2}, i \in I\big) = \arg\min J_{\text{NEN}}(\mathbf{a}|\lambda_1, \lambda_2) \in \mathbb{R}^n. \end{aligned} \tag{4}$$

In contrast to the "naive" Elastic Net, an improved training criterion is proposed in [1] as the "proper" Elastic Net, which may be straightforwardly shown to differ from (3) only by the quadratic penalty term:

$$J_{\mathrm{EN}}(\mathbf{a}|\lambda_1,\lambda_2) = \lambda_2 \sum_{i=1}^{n}(a_i - a_i^*)^2 + \lambda_1 \sum_{i=1}^{n}|a_i| + \sum_{j=1}^{N}\Big(y_j - \sum_{i=1}^{n} a_i x_{ij}\Big)^2$$

$$= \lambda_2\Big(\mathbf{a} - \frac{1}{N}\mathbf{X}^T\mathbf{y}\Big)^T\Big(\mathbf{a} - \frac{1}{N}\mathbf{X}^T\mathbf{y}\Big) + \lambda_1\|\mathbf{a}\|_1 + \big(\mathbf{y} - \mathbf{X}\mathbf{a}\big)^T\big(\mathbf{y} - \mathbf{X}\mathbf{a}\big) \to \min(\mathbf{a}), \tag{5}$$

$$\hat{\mathbf{a}}_{\lambda_1,\lambda_2} = \big(\hat{a}_{i,\lambda_1,\lambda_2}, i \in I\big) = \arg\min J_{\mathrm{EN}}(\mathbf{a}|\lambda_1,\lambda_2) \in \mathbb{R}^n, \tag{6}$$

where $\mathbf{a}^* = (1/N)\mathbf{X}^T\mathbf{y}$ is vector of preliminary independent estimates of regression coefficients derived from the normalized training data (4) as a set of observed covariances

$$\mathbf{a}^* = \Big(a_i^* = \frac{1}{N}\sum_{j=1}^{N} y_j x_{ij},\ i = 1,\ldots,n\Big) = \frac{1}{N}\mathbf{X}^T\mathbf{y} \in \mathbb{R}^n. \tag{7}$$

Use of bias within the quadratic regularization term is motivated in [1] by the intention of decorrelating the feature vectors in the training set $(\mathbf{x}_j, j = 1,\ldots,N)$. However, what is actually analyzed in [1] is the lasso-like form of (5):

$$\frac{\lambda_1}{1+\lambda_2/N}\|\mathbf{a}\|_1 + \left[\mathbf{a}^T\frac{\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2/N}\mathbf{a} - 2\mathbf{y}^T\mathbf{X}\mathbf{a}\right] \to \min(\mathbf{a}). \tag{8}$$

Theorem 1. *The training criteria* (5) *and* (8) *are equivalent.*

(**Proof** is given in Appendix A[1]).

It is clear that the "naive" Elastic Net (3) is a special case of (5) with $\mathbf{a}^* = \mathbf{0}$, just as the Lasso criterion is a further special case with $\lambda_2 = 0$.

In order to tune the structural parameters λ_1 and λ_2, tenfold cross-validation is applied in [1], since determining the more reliable leave-one-out error rate proves to be computationally too expensive for large training sets.

In this paper, we propose to retain the full leave-one-out procedure omitted by Zou and Hastie [1] without, however, multiplying the computational complexity of the training procedure. To do this, we exploit the inherent capacity of the Elastic Net training criterion (5) to partition the set of input variables into three subsets defined by negative, positive, and zero values of their corresponding regression coefficients.

We thus, in Sect. 2, treat the input-variable partitioning at the minimum of the Elastic Net criterion as a *secondary regularization parameter* produced by the *primary parameters* λ_1 and λ_2, one which completely determines the variable

[1] In [1], denominators in (5) have the form $1 + \lambda_2$ instead of $1 + \lambda_2/N$. This is a consequence of a specific normalization of the training set $\sum_{j=1}^{N} x_{ij}^2 = 1$ as distinct to the commonly adopted normalization $(1/N)\sum_{j=1}^{N} x_{ij}^2 = 1$ accepted in this paper (2).

selection. The resulting partition enforces a strictly quadratic Elastic Net criterion with respect to the active regression coefficients.

In Sect. 3, the latter property allows for non-enumerative computation of the leave-one-out error rate, thereby avoiding the multiple training repetitions that would otherwise be required to determine model generality for tuning the structural parameters. This approach is well known in mathematical statistics [7] but needs detailed elaboration when applied to the Elastic Net.

Finally, the results of a simulation study are presented in Sect. 4. The proposed methodology is verified in the same ground-truth experimental framework that was used by Zou and Hastie in their original paper on the Elastic net [1].

2 Optimal Partitioning of the Set of Regression Coefficients: A Secondary Non-numeric Structural Parameter

Let $\{(\mathbf{x}_j, y_j), j = 1, \ldots, N\}$ be the training set, centered, and normalized in accordance with (1). Let, further, $I = \{1, \ldots, n\}$ be the set of indices of real-valued features $x_i \in \mathbb{R}$, $i \in I$, assigned to each entity, so that $x_{ij} \in \mathbb{R}$. The Elastic Net training criterion (5) is a convex function $J_{\mathrm{EN}}(\mathbf{a}|\lambda_1, \lambda_2) : \mathbb{R}^N \to \mathbb{R}$, whose minimum point $\hat{\mathbf{a}}_{\lambda_1,\lambda_2} = (\hat{a}_{i,\lambda_1,\lambda_2}, i \in I)$ (6) is the vector of regression coefficients to be inferred from the training set.

It is shown in [1] that an intrinsic property of the Elastic Net at its minimum is a natural partitioning of the feature set $I = \{1, \ldots, n\}$ into three nonintersecting subsets associated with negative, positive, and strictly zero values of the estimated regression coefficients:

$$\begin{cases} \hat{I}^-_{\lambda_1,\lambda_2} = \{i \in I : \hat{a}_{i,\lambda_1,\lambda_2} < 0\}, \\ \hat{I}^0_{\lambda_1,\lambda_2} = \{i \in I : \hat{a}_{i,\lambda_1,\lambda_2} = 0\}, \; I = \hat{I}^-_{\lambda_1,\lambda_2} \bigcup \hat{I}^0_{\lambda_1,\lambda_2} \bigcup \hat{I}^+_{\lambda_1,\lambda_2}. \\ \hat{I}^+_{\lambda_1,\lambda_2} = \{i \in I : \hat{a}_{i,\lambda_1,\lambda_2} > 0\}, \end{cases} \tag{9}$$

In the following, we shall use the notations

$$\begin{aligned} \hat{n}_{\lambda_1,\lambda_2} &= n - |\hat{I}^0_{\lambda_1,\lambda_2}| = |\hat{I}^-_{\lambda_1,\lambda_2}| + |\hat{I}^+_{\lambda_1,\lambda_2}|, \\ \hat{n}^0_{\lambda_1,\lambda_2} &= |\hat{I}^0_{\lambda_1,\lambda_2}|, \;\; \hat{n}^-_{\lambda_1,\lambda_2} = |\hat{I}^-_{\lambda_1,\lambda_2}|, \;\; \hat{n}^+_{\lambda_1,\lambda_2} = |\hat{I}^+_{\lambda_1,\lambda_2}|, \\ n &= \hat{n}^0_{\lambda_1,\lambda_2} + \hat{n}_{\lambda_1,\lambda_2} = \hat{n}^0_{\lambda_1,\lambda_2} + \hat{n}^-_{\lambda_1,\lambda_2} + \hat{n}^+_{\lambda_1,\lambda_2}, \end{aligned} \tag{10}$$

to denote, as appropriate, the numbers of zero-valued, negative, and positive regression coefficients, and more generally, the total number of passive and active regressors, determined by the partition (9). This partition is an integral part of the output produced, for instance, by the well-known algorithm LARS-EN [1], developed specifically for solving the Elastic Net problem defined in (5) as a generalized version of the LARS algorithm previously developed for the Lasso problem with $\lambda_2 = 0$ [8].

The particular subset of $\hat{n}_{\lambda_1,\lambda_2}$ active (i.e., nonzero) regression coefficients arrived at, i.e. $\hat{I}^-_{\lambda_1,\lambda_2} \bigcup \hat{I}^+_{\lambda_1,\lambda_2} \subseteq I$, thus explicitly manifests the principal aim of the Elastic Net regularization, namely the selection of informative features and the suppression of redundant ones. Since the partition (9) is explicitly tied to the Elastic Net parameters λ_1 and λ_2, it would appear natural to consider it as a secondary non-numeric structural parameter of the regression estimation.

Having been specified, the resulting partition (9) along with the primary structural parameters (λ_1, λ_2) jointly make the Elastic Net criterion (5) strictly quadratic with respect to the active regression coefficients:

$$J_{\mathrm{EN}}\left(a_i, i \notin \hat{I}^0_{\lambda_1,\lambda_2} | \lambda_1, \lambda_2\right) = \lambda_2 \sum_{i \notin \hat{I}^0_{\lambda_1,\lambda_2}} \left(a_i - a_i^*\right)^2 - \lambda_1 \sum_{i \in \hat{I}^-_{\lambda_1,\lambda_2}} a_i + \lambda_1 \sum_{i \in \hat{I}^+_{\lambda_1,\lambda_2}} a_i$$

$$+ \sum_{j=1}^{N} \left(y_j - \sum_{i \notin \hat{I}^0_{\lambda_1,\lambda_2}} a_i x_{ij}\right)^2 \to \min\left(a_i, i \notin \hat{I}^0_{\lambda_1,\lambda_2}\right), \quad a_i^* = \frac{1}{N} \sum_{j=1}^{N} y_j x_{ij}. \quad (11)$$

It will be convenient to introduce the following notation for the two subvectors and one submatrix (corresponding to two vectors and one matrix $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{x}_j \in \mathbb{R}^n$ and $\mathbf{X}(N \times n)$ (4)) "cut out" by the formation of the partition:

$$\tilde{\mathbf{a}}_{\lambda_1,\lambda_2} = \left(a_i, i \notin \hat{I}^0_{\lambda_1,\lambda_2}\right) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}, \; \tilde{\mathbf{x}}_{j,\lambda_1,\lambda_2} = \left(x_{ij}, i \notin \hat{I}^0_{\lambda_1,\lambda_2}\right) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}},$$

$$\tilde{\mathbf{X}}_{\lambda_1,\lambda_2} = \left(\tilde{\mathbf{x}}_1 \cdots \tilde{\mathbf{x}}_N\right)^T (N \times \hat{n}_{\lambda_1,\lambda_2}).$$

In addition, special notation will be required for the vector indicating membership of regression features in subsets $\hat{I}^-_{\lambda_1,\lambda_2}$ and $\hat{I}^+_{\lambda_1,\lambda_2}$

$$\tilde{\mathbf{e}}_{\lambda_1,\lambda_2} = \left(e_i, i \notin \hat{I}^0_{\lambda_1,\lambda_2}\right) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}, \; \tilde{e}_i = \begin{cases} +1, \; i \in \hat{I}^+_{\lambda_1,\lambda_2}, \\ -1, \; i \in \hat{I}^-_{\lambda_1,\lambda_2}, \end{cases}$$

as well as for the subvector cut out of $\mathbf{a}^*$ (7):

$$\tilde{\mathbf{a}}^*_{\lambda_1,\lambda_2} = \left(a_i^*, i \notin \hat{I}^0_{\lambda_1,\lambda_2}\right) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}. \quad (12)$$

Theorem 2. *The solution* $\hat{\mathbf{a}}_{\lambda_1,\lambda_2} = \left(\hat{a}_{i,\lambda_1,\lambda_2}, i \in I\right) \in \mathbb{R}^n$ *of the Elastic Net training problem* (5) *is a combination of the solution* $\hat{\tilde{\mathbf{a}}}_{\lambda_1,\lambda_2} = \left(\hat{a}_{i,\lambda_1,\lambda_2}, i \notin \hat{I}^0_{\lambda_1,\lambda_2}\right) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}$ *of* (11) *with respect to the partition* (9) *and equalities* $\left(\hat{a}_{i,\lambda_1,\lambda_2} = 0, i \in \hat{I}^0_{\lambda_1,\lambda_2}\right)$. *In turn, vector* $\hat{\tilde{\mathbf{a}}}_{\lambda_1,\lambda_2}$ *is a solution*

$$\hat{\tilde{\mathbf{a}}}_{\lambda_1,\lambda_2} = \left(\tilde{\mathbf{X}}^T_{\lambda_1,\lambda_2} \tilde{\mathbf{X}}_{\lambda_1,\lambda_2} + \lambda_2 \tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}}\right)^{-1} \left[\tilde{\mathbf{X}}^T_{\lambda_1,\lambda_2} \mathbf{y} - \frac{\lambda_1}{2} \tilde{\mathbf{e}}_{\lambda_1,\lambda_2} + \lambda_2 \tilde{\mathbf{a}}^*\right] \quad (13)$$

of the system of $\hat{n}_{\lambda_1,\lambda_2}$ linear equations over the same number of variables:

$$\left(\tilde{\mathbf{X}}^T_{\lambda_1,\lambda_2}\tilde{\mathbf{X}}_{\lambda_1,\lambda_2} + \lambda_2\tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}}\right)\tilde{\mathbf{a}} = \tilde{\mathbf{X}}^T_{\lambda_1,\lambda_2}\mathbf{y} - \frac{\lambda_1}{2}\tilde{\mathbf{e}}_{\lambda_1,\lambda_2} + \lambda_2\tilde{\mathbf{a}}^*. \tag{14}$$

(**Proof** is given in Appendix B).

It would not, in itself, make sense to directly solve the equation system (14) for estimating the active regression coefficients once again, because the full set of estimates $\hat{\mathbf{a}}_{\lambda_1,\lambda_2} = \left(\hat{a}_i, i \in I\right) \in \mathbb{R}^n$ can be found by any appropriate algorithm for minimizing the convex function (5). However, the format of Theorem 2 suggests the possibility of considering the optimal feature partition of the set of regression coefficients as constituting just that structural parameter associated with (λ_1, λ_2) which is to be verified by the leave-one-out criterion.

The larger the subset of excluded features $\hat{I}^0_{\lambda_1,\lambda_2} \subseteq I$, the lower the complexity of the class of regression models expressed by criterion (11). In particular, the LARS-EN algorithm of [1] explicitly yields the feature partitioning induced by the primary parameters (λ_1, λ_2). Thus, this partitioning may serve as the secondary structural parameter of the regression model, one that quantitatively acts a proxy for the overall model complexity.

3 Non-enumerative Leave-One-Out Verification of the Structural Parameters

3.1 Leave-One-Out Verification of the Feature Partitioning

We will assume that the Elastic Net problem (5) has been solved for the given training set (1) at certain values of structural parameters (λ_1, λ_2), and that estimates of regression coefficients $\hat{\mathbf{a}}_{\lambda_1,\lambda_2} = \left(\hat{a}_i, i \in I\right)$ (6) along with the feature partition (9) have been found. The corresponding average least squares residual is given by:

$$\hat{S}(\lambda_1, \lambda_2) = \frac{1}{N}\sum_{j=1}^{N}\hat{\delta}^2_{j,\lambda_1,\lambda_2}, \tag{15}$$

$$\hat{\delta}_{j,\lambda_1,\lambda_2} = y_j - \sum_{i \notin \hat{I}^0_{\lambda_1,\lambda_2}} \hat{a}_{i,\lambda_1,\lambda_2}x_{ij} = y_j - \tilde{\mathbf{x}}^T_j\hat{\tilde{\mathbf{a}}}_{\lambda_1,\lambda_2} = y_j - \hat{y}_{j,\lambda_1,\lambda_2}. \tag{16}$$

As applied to the hypothetical training criterion (11) regularized by the structural parameters (λ_1, λ_2) with the related feature partition (9), leave-one-out verification consists, generally speaking, in an N-fold execution of the following steps:

- delete one entity, say the k th feature vector $\mathbf{x}_k$, from the training set (1), and recompute the vector of preliminary estimates in (11) and (12):

$a_i^{*(k)} = \big(1/(N-1)\big) \sum_{j=1, j\neq k} y_j x_{ij}$, $\tilde{\mathbf{a}}^{*(k)} = \big(a_i^{*(k)}, i \notin \hat{I}^0\big) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}$;

- estimate the regression coefficients in accordance with $\sum_{j=1, j\neq k}^{N} (y_j - \ldots)^2$ in (11) from the remaining set of entities $\hat{\tilde{\mathbf{a}}}_{\lambda_1,\lambda_2}^{(k)} = \big(\hat{a}_{i,\lambda_1,\lambda_2}^{(k)}, i \notin \hat{I}_{\lambda_1,\lambda_2}^0\big) \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}$;
- compute the prediction error at the deleted entity $\hat{\delta}_{k,\lambda_1,\lambda_2}^{(k)} = y_k - \hat{y}_{k,\lambda_1,\lambda_2}^{(k)}$.

Finally, average the squared errors over the entire training set $k = 1, \ldots, N$.

The resulting leave-one-out rate $\hat{S}_{\mathrm{LOO}}(\lambda_1, \lambda_2)$, in contrast to (15), constitutes the average risk estimate computed from the training set available to the observer:

$$\hat{S}_{\mathrm{LOO}}(\lambda_1, \lambda_2) = \frac{1}{N} \sum_{k=1}^{N} (\hat{\delta}_{k,\lambda_1,\lambda_2}^{(k)})^2, \tag{17}$$

$$\hat{\delta}_{k,\lambda_1,\lambda_2}^{(k)} = y_k - \hat{y}_{k,\lambda_1,\lambda_2}^{(k)} = y_k - \tilde{\mathbf{x}}_k^T \hat{\tilde{\mathbf{a}}}_{\lambda_1,\lambda_2}^{(k)}. \tag{18}$$

It should be noted that deletion of one entity from the training set (1) potentially destroys centering and normalization (2). Generally speaking, recentering and renormalizing is therefore required before computing each leave-one-out residual $\hat{\delta}_{k,\lambda_1,\lambda_2}^{(k)}$ in (17) for maximal performance. However, we omit these operations for the sake of simplicity.

3.2 The Efficient Leave-One-Out Procedure

At first glance, it would appear that computing each leave-one-out residual (18) would require a separate instantiation of the solution to the problem (11) with $\sum_{j=1, j\neq k}^{N} (y_j - \ldots)^2$. Fortunately, however, the quadratic form of criterion (11) allows us to avoid multiple optimizations when computing the leave-one-out error rate (17). The principle we shall employ for rapid computation of the leave-one-out error rate for quadratic training criteria is given in [7]. The aim of the current paper is thus to adapt this approach to the particular case of Elastic Net regression regularization.

The following theorem demonstrates that each leave-one-out residual $\hat{\delta}_{k,\lambda_1,\lambda_2}^{(k)}$ (18) in (17) can be easily computed from the respective residual $\hat{\delta}_{k,\lambda_1,\lambda_2}$ estimated from the entire training set (16).

Theorem 3. *Assume that the Elastic Net problem* (5) *has been solved for the entire training set with structural parameters* λ_1 *and* λ_2*, i.e., the smallest residuals* $\hat{\delta}_{j,\lambda_1,\lambda_2}$ (16) *have been found along with the feature partition* $I = \hat{I}_{\lambda_1,\lambda_2}^- \bigcup \hat{I}_{\lambda_1,\lambda_2}^0 \bigcup \hat{I}_{\lambda_1,\lambda_2}^+$ (9)*. Then the leave-one-out rate* (17) *allows the following representations for, respectively, the "naive"* (3) *and "proper" Elastic Net* (5) *formulations:*

$$\hat{S}_{\mathrm{LOO}}^{\mathrm{NEN}}(\lambda_1,\lambda_2)=\frac{1}{N}\sum_{k=1}^{N}\left(\frac{\hat{\delta}_{k,\lambda_1,\lambda_2}}{1-q_{k,\lambda_1,\lambda_2}}\right)^2 \ (\textit{NaiveElasticNet}), \tag{19}$$

$$\hat{S}_{\mathrm{LOO}}^{\mathrm{EN}}(\lambda_1,\lambda_2)=\frac{1}{N}\sum_{k=1}^{N}\left(\frac{\hat{\delta}_{k,\lambda_1,\lambda_2}+\frac{1}{N-1}\lambda_2\left(y_k q_{k,\lambda_1,\lambda_2}-h_{k,\lambda_1,\lambda_2}\right)}{1-q_{k,\lambda_1,\lambda_2}}\right)^2 \ (\textit{ElasticNet}), \tag{20}$$

where $\tilde{\mathbf{a}}^*$ *is the initial preliminary estimate of the regression coefficients over the entire training set* (7)*,*

$$\begin{aligned} q_{k,\lambda_1,\lambda_2} &= \tilde{\mathbf{x}}_k^T\left(\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}^T\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}+\lambda_2\tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}}\right)^{-1}\tilde{\mathbf{x}}_k,\\ h_{k,\lambda_1,\lambda_2} &= \tilde{\mathbf{x}}_k^T\left(\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}^T\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}+\lambda_2\tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}}\right)^{-1}\tilde{\mathbf{a}}^*. \end{aligned} \tag{21}$$

(**Proof** is given in Appendix C).

It may be seen from (21) that the inverse matrix $\left(\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}^T\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}+\lambda_2\tilde{\mathbf{I}}_{\hat{n}_{\lambda_1,\lambda_2}}\right)^{-1}$ is computed only once when estimating the regression coefficients over the entire training set (13); furthermore, it remains the same for all $k=1,\ldots,N$, from which the efficiency of our method derives.

4 Experimental Study with Simulated Data

We illustrate the operation of both versions of our non-enumerative leave-one-out procedure (19) and (20) with the synthetic data used by Zou and Hastie in their original paper [1] in order to demonstrate the efficiency of their method with respect to standard Lasso.

In the same manner as Zou and Hastie, we thus randomly simulate data sets $\{(\mathbf{x}_j, y_j), j=1,\ldots,N\}$ from the ground-truth model:

$$\mathbf{y}=\mathbf{X}\mathbf{a}+\sigma\boldsymbol{\varepsilon},\ \mathbf{y},\boldsymbol{\varepsilon}\in\mathbb{R}^N,\ \mathbf{a}\in\mathbb{R}^n,\ \ \boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}), \tag{22}$$

where $\mathbf{X}=\left(\mathbf{x}_1\cdots\mathbf{x}_N\right)^T$ $(N\times n)$ is a sample of independent random vectors $\mathbf{x}_j=(x_{1j}\cdots x_{nj})^T\in\mathbb{R}^n$ normally distributed in accordance with the covariance matrix $\left[\mathrm{Cov}(i,l),\ i,l=1,\ldots,n\right]$, $Cov(i,i)=1$.

As in [1], four experimental examples are selected; however, certain necessary differences occur due to the use of leave-one-out verification. In the original paper, the simulated data within each example consisted of a training set, an independent validation set, and an independent test set with respective magnitudes $N_{\mathrm{tr}}/N_{\mathrm{val}}/N_{\mathrm{Test}}$, with the initial training set only once divided into the two subsets used for training and validation. In contrast to [1], we apply leave-one-out

cross-validation, i.e. executing as many divisions as the number of training entities. Thus, the simulated data set within each of our experiments consists of a training set and an independent test set with respective magnitudes $N_{\mathrm{Tr}}/N_{\mathrm{Test}}$, where $N_{\mathrm{Tr}} = N_{\mathrm{tr}} + N_{\mathrm{val}}$.

Other than the above, the details of the four example scenarios are the same as in [1]:

(1) In Example 1, we simulate 50 data sets consisting of 40/200 observations, instead of 20/20/200 as in [1], and employ 8 predictors: $\mathbf{x}_j = (x_{1j} \cdots x_{8j})$, $n = 8$. We let

$$\mathbf{a} = \big(3.0,\ 1.5,\ 0.0,\ 0.0,\ 2.0,\ 0.0,\ 0.0,\ 0.0\big) \in \mathbb{R}^8.$$

The covariance between x_i and x_l is given by $\mathrm{Cov}(i, l) = 0.5^{|i-l|}$.

(2) Example 2 is identical to Example 1, except that $a_i = 0.85$ for all i.

(3) In Example 3, we simulate 50 data sets consisting of 200/400 observations, instead of 100/100/400 as in [1], and employ 40 predictors $\mathbf{x}_j = (x_{1j} \cdots x_{40j})$, $n = 40$. We set

$$\mathbf{a} = \big(\underbrace{0.0, \ldots, 0.0}_{10}, \underbrace{2.0, \ldots, 2.0}_{10}, \underbrace{0.0, \ldots, 0.0}_{10}, \underbrace{2.0, \ldots, 2.0}_{10}\big) \in \mathbb{R}^{40},$$

$\sigma = 15$, and $\mathrm{Cov}(i, l) = 0.5$ for all i and l.

(4) In Example 4, we simulate 50 data sets consisting of 100/400 observations, instead of 50/50/400 in [1], and 40 predictors. We choose

$$\mathbf{a} = \big(\underbrace{3.0, \ldots, 3.0}_{15}, \underbrace{0.0, \ldots, 0.0}_{25}\big) \in \mathbb{R}^{40},$$

and $\sigma = 15$. The predictors $\mathbf{x} = \big(x_1 \cdots x_{40}\big)$ were generated as follows:

$$\left.\begin{array}{l} x_i = z_1 + \varepsilon_i^x,\ z_1 \sim \mathcal{N}(0,1),\ i = 1, \ldots, 5, \\ x_i = z_2 + \varepsilon_i^x,\ z_2 \sim \mathcal{N}(0,1),\ i = 6, \ldots, 10, \\ x_i = z_3 + \varepsilon_i^x,\ z_3 \sim \mathcal{N}(0,1),\ i = 11, \ldots, 15, \\ x_i \sim \mathcal{N}(0,1),\ \text{i.i.d.},\ i = 16, \ldots, 40. \end{array}\right\} \varepsilon_i^x \sim \mathcal{N}(0,\ 0.01),\ \text{i.i.d.},$$

This model consists of three equally important groups each containing five members, and, additionally, 25 pure noise features.

For each of the 50 random data sets in each of the four examples, we twice solve the Naive Elastic Net and Elastic Net problems (3) and (5) using the versions of LARS-EN available on the sites
http://www-stat.stanford.edu/~tibs/glmnet-matlab/ for naive Elastic Net and
http://cran.r-project.org/web/packages/elasticnet/index.html for Elastic net.

Table 1 Median root-mean-square test error based on 50 replications for the four methods

Method of regression estimation/cross-validated choice of structural parameters	Median root-mean-square test error, percent			
	Example 1	Example 2	Example 3	Example 4
Naive Elastic Net, onefold cross validation	3.47	3.40	16.80	24.21
Naive Elastic Net, leave-one-out cross validation	3.33	3.29	16.79	19.20
Elastic Net, onefold cross validation	3.44	3.44	17.45	24.21
Elastic Net, leave-one-out cross validation	3.33	3.29	16.79	19.20

At each run of the program, the regularization parameter λ_2 was set to be constant. As to the parameter λ_1, its $n + 1$ tentative values, where n is the full number of variables in the data model (22), were produced by the regularization path inbuilt in the program. The resulting decrement in the values of λ_1 determines the respective succession of $n + 1$ feature partitionings (9)–(10), starting with $\hat{I}^0_{\lambda_1,\lambda_2} = I$, $\hat{n}_{\lambda_1,\lambda_2} = 0$, and ending with $\hat{I}^0_{\lambda_1,\lambda_2} = \varnothing$, $\hat{n}_{\lambda_1,\lambda_2} = n$. Additionally, we varied the preset structural parameter λ_2.

In the first experimental phase, this procedure was applied to the unified training set of magnitude $N_{\text{Tr}} = N_{\text{tr}} + N_{\text{val}}$, the structural parameters (λ_1, λ_2) were chosen as the values providing the minimum value of the quick leave-one-out indicator (19) or (20), and the mean-square error was computed over the test set of size N_{Test}.

In the second phase, the same procedure was applied to the initial training set of half size, i.e. $N_{\text{tr}} = N_{\text{Tr}}/2$, and the structural parameters were derived by minimization of the error over the validation set of the same size $N_{\text{val}} = N_{\text{Tr}}/2$, i.e., in accordance with the onefold cross validation principle, just as was done in [1]. The final error rate was computed on the test set.

Table 1 summarizes the prediction results in the above four examples, averaged over all the 50 random data sets. It can be seen, as expected, that the leave-one-out verification of tentative pairs (λ_1, λ_2) provides a better choice of structural parameters in terms of the mean-square error rate on the test set than the onefold cross validation.

5 Conclusion

We propose, in this paper, a computationally efficient non-enumerative algorithm for computation of the leave-one-out error rate in Zou and Hastie's Elastic Net regularization [1], one which enables determination of model generality for tuning structural parameters in situ while avoiding multiple training repetitions. To do so, we consider the partitioning of features at the minimum of the Elastic Net criterion as a secondary regularization parameter, such that the resulting partition comprises a strictly quadratic Elastic Net criterion.

The proposed methodology is applied to the ground-truth experimental framework used by Zou and Hastie in their original paper [1]. We determine that

the accuracy of the two methods is essentially identical, with a slight advantage for the leave-one-out verification. However, the computation time is significantly reduced by the explicit incorporation of the non-enumerative leave-one-out error rate calculation.

References

1. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. Roy. Stat. Soc. **67**, 301–320 (2005)
2. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. **58**(1), 267–288 (1996)
3. Ye, G., Chen, Y., Xie, X.: Efficient variable selection in support vector machines via the alternating direction method of multipliers. J. Mach. Learn. Res. Proc. Track 832–840 (2011)
4. Wang, L., Zhu, J., Zou, H.: The doubly regularized support vector machine. Stat. Sinica **16**, 589–615 (2006)
5. Grosswindhager, S.: Using penalized logistic regression models for predicting the effects of advertising material (2009). http://publik.tuwien.ac.at/files/PubDat_179921.pdf
6. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**, 1–22 (2010)
7. Christensen, R.: Plane Answers to Complex Questions. The Theory of Linear Models, 3rd edn. Springer, New York (2010)
8. Tibshirani, R., Efron, B., Hastie, T., Johnstone, I.: Least angle regression. Ann. Stat. **32**, 407–499 (2004)

Appendix

Proof of Theorem 1

Let us open out the brackets in (5):

$$J_{\mathrm{EN}}(\mathbf{a}|\lambda_1,\lambda_2) = \lambda_1\|\mathbf{a}\|_1 + \lambda_2\mathbf{a}^T\mathbf{a} - 2\frac{\lambda_2}{N}\mathbf{a}^T\mathbf{X}^T\mathbf{y} + \underbrace{\frac{\lambda_2}{N^2}\mathbf{y}^T\mathbf{X}\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}}_{\text{const}} - 2\mathbf{a}^T\mathbf{X}^T\mathbf{y} + \mathbf{a}^T\mathbf{X}^T\mathbf{X}\mathbf{a} \to \min(\mathbf{a}).$$

Summands not depending on $\mathbf{a}$ may be omitted from the optimization. Collecting the remaining summands gives:

$$J_{\mathrm{EN}}(\mathbf{a}|\lambda_1,\lambda_2) = \lambda_1\|\mathbf{a}\|_1 + \mathbf{a}^T\left(\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}\right)\mathbf{a} + \left(1 + \frac{\lambda_2}{N}\right)\mathbf{a}^T\mathbf{X}^T\mathbf{y} \to \min(\mathbf{a}).$$

Division of the last equality by the constant $(1 + \lambda_2/N)$ yields (8). **The theorem is proven**.

Proof of Theorem 2

Differentiation of (11) by the active regression coefficients a_i, $i \notin \hat{I}^0_{\lambda_1,\lambda_2}$, leads to the equalities

$$\frac{\partial}{\partial a_i} J_{\mathrm{EN}}\big(a_l, l \notin \hat{I}^0_{\lambda_1,\lambda_2} | \lambda_1, \lambda_2\big)$$

$$= 2\lambda_2(a_i - a_i^*)^2 + \begin{pmatrix} \lambda_1, i \in \hat{I}^+_{\lambda_1,\lambda_2} \\ -\lambda_1, i \in \hat{I}^-_{\lambda_1,\lambda_2} \end{pmatrix} - 2\sum_{j=1}^{N}\Big(y_j - \sum_{l \notin \hat{I}^0_{\lambda_1,\lambda_2}} a_l x_{lj}\Big) = 0,$$

which make a system of linear equations over $i \notin \hat{I}^0_{\lambda_1,\lambda_2}$

$$\lambda_2 a_i + \sum_{l \notin \hat{I}^0_{\lambda_1,\lambda_2}} \left(\sum_{j=1}^{N} x_{ij} x_{lj}\right) a_l = \sum_{j=1}^{N} x_{ij} y_j - \frac{\lambda_1}{2} \begin{pmatrix} 1, i \in \hat{I}^+_{\lambda_1,\lambda_2} \\ -1, i \in \hat{I}^-_{\lambda_1,\lambda_2} \end{pmatrix} + \lambda_2 \mathbf{a}^*.$$

The matrix form of this system in accordance with (12), (13), and (14) is just (16), with (13) its solution. **The theorem is proven**.

Proof of Theorem 3

Let the feature set partitioning $\{\hat{I}^-_{\lambda_1,\lambda_2}, \hat{I}^0_{\lambda_1,\lambda_2}, \hat{I}^+_{\lambda_1,\lambda_2}\}$ (9) at the minimum point of (5) be treated as fixed, and the k th entity $(\mathbf{x}_k, y_k)$ be omitted from the training set (1). In terms of notation (4) and (2), this implies deletion of the k element from the vector $\mathbf{y} \in \mathbb{R}^N$ and the kth row from the matrix $\tilde{\mathbf{X}}_{\lambda_1,\lambda_2}$ $(N \times \hat{n}_{\lambda_1,\lambda_2})$:

$$\mathbf{y}^{(k)} \in \mathbb{R}^{N-1},\ \tilde{\mathbf{X}}^{(k)}_{\lambda_1,\lambda_2}\big((N-1) \times \hat{n}_{\lambda_1,\lambda_2}\big).$$

The vector of preliminary estimates of regression coefficients $\mathbf{a}^* \in \mathbb{R}^n$ (12) occurs only in the Elastic Net (EN) training criterion (5), and equals zero in the naive Elastic Net (NEN) (3) $\mathbf{a}^* = \mathbf{0} \in \mathbb{R}^n$. Its subvector cut out from $\mathbf{a}^*$ by deletion of the kth entity will be:

$$\tilde{\mathbf{a}}^{*(k)}_{\lambda_1,\lambda_2} = \begin{cases} \dfrac{1}{N-1} \displaystyle\sum_{j=1, j\neq k}^{N} y_j \tilde{\mathbf{x}}_{j,\lambda_1,\lambda_2} = \frac{1}{N-1}(\tilde{\mathbf{X}}^{(k)}_{\lambda_1,\lambda_2})^T \mathbf{y}^{(k)} \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}, & EN \\ \mathbf{0} \in \mathbb{R}^{\hat{n}_{\lambda_1,\lambda_2}}, & NEN \end{cases}$$

Correspondingly, the solution (13) of the optimization problem (11) will take the form (lower indices (λ_1, λ_2) are omitted below):

$$\hat{\tilde{\mathbf{a}}}^{(k)} = \left((\tilde{\mathbf{X}}^{(k)})^T \tilde{\mathbf{X}}^{(k)} + \lambda_2 \tilde{\mathbf{I}}_{\hat{n}}\right)^{-1} \left\{ (\tilde{\mathbf{X}}^{(k)})^T \mathbf{y} - \frac{\lambda_1}{2} \tilde{\mathbf{e}} + \begin{bmatrix} \lambda_2 \tilde{\mathbf{a}}^{*(k)}, & EN \\ \mathbf{0}, & NEN \end{bmatrix} \right\}. \tag{23}$$

Notice here that

$$\begin{cases} (\tilde{\mathbf{X}}^{(k)})^T \tilde{\mathbf{X}}^{(k)} = \tilde{\mathbf{X}}^T \tilde{\mathbf{X}} - \tilde{\mathbf{x}}_k^T \tilde{\mathbf{x}}_k, \\ (\tilde{\mathbf{X}}^{(k)})^T \mathbf{y}^{(k)} = \tilde{\mathbf{X}}^T \mathbf{y} - y_k \tilde{\mathbf{x}}_k, \\ \tilde{\mathbf{a}}^{*(k)} = \dfrac{1}{N-1}\left[\tilde{\mathbf{X}}^T \mathbf{y} - y_k \tilde{\mathbf{x}}_k\right] = \dfrac{N}{N-1} \tilde{\mathbf{a}}^* - \dfrac{1}{N-1} y_k \tilde{\mathbf{x}}_k \\ = \tilde{\mathbf{a}}^* - \dfrac{1}{N-1}\left(y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*\right). \end{cases} \tag{24}$$

Application of the Woodbury formula[1]

$$(\mathbf{A} + \mathbf{BC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B}\right)^{-1}\mathbf{CA}^{-1}$$

and (24) to (23) yields:

$$\begin{aligned} \hat{\tilde{\mathbf{a}}}^{(k)} &= \Big(\underbrace{\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}}_{\mathbf{A}} + \underbrace{(-\tilde{\mathbf{x}}_k)}_{\mathbf{B}} \underbrace{\tilde{\mathbf{x}}_k^T}_{\mathbf{C}}\Big)^{-1} \\ &\quad \times \left\{ \tilde{\mathbf{X}}^T \mathbf{y} - y_k \tilde{\mathbf{x}}_k - \frac{\lambda_1}{2} \tilde{\mathbf{e}} + \lambda_2 \begin{bmatrix} \tilde{\mathbf{a}}^* - \dfrac{1}{N-1}\left(y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*\right), & EN \\ \mathbf{0}, & NEN \end{bmatrix} \right\} \\ &= \hat{\tilde{\mathbf{a}}} + \frac{\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T \hat{\tilde{\mathbf{a}}}}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} - \frac{y_k}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k \\ &\quad - \frac{\lambda_2}{N-1} \begin{bmatrix} \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} + \dfrac{\left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1}}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} \left(y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*\right), & EN \\ \mathbf{0}, & NEN \end{bmatrix}. \end{aligned}$$

Algebraic transformation of this expression with respect to the notation $\hat{y}_k = \tilde{\mathbf{x}}_k^T \hat{\tilde{\mathbf{a}}}$ (16) and $\hat{y}_k^{(k)} = \tilde{\mathbf{x}}_k^T \hat{\tilde{\mathbf{a}}}^{(k)}$ (18) leads to the equality

$$\begin{aligned} \tilde{\mathbf{x}}_k^T \hat{\tilde{\mathbf{a}}}^{(k)} &= \frac{\hat{y}_k}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} - y_k \frac{\tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} \\ &\quad - \frac{\lambda_2}{N-1} \begin{bmatrix} \dfrac{\tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \left(y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*\right)}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k}, & EN \\ \mathbf{0}, & NEN \end{bmatrix}. \end{aligned}$$

[1] http://en.wikipedia.org/wiki/Woodbury_matrix_identity.

Thus, the leave-one-out residuals $\hat{\delta}_k^{(k)}$ in (17) and (18) permit the representation

$$\begin{aligned}
\hat{\delta}_k^{(k)} &= y_k - \tilde{\mathbf{x}}_k^T \hat{\tilde{\mathbf{a}}}^{(k)} \\
&= y_k - \frac{\hat{y}_k}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} - y_k \frac{\tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} \\
&\quad - \frac{\lambda_2}{N-1} \begin{bmatrix} \dfrac{\tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} (y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*)}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k}, & EN \\ \mathbf{0}, & NEN \end{bmatrix} \\
&= \frac{y_k - \hat{y}_k}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k} \\
&\quad + \frac{\lambda_2}{N-1} \begin{bmatrix} \dfrac{\tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} (y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*)}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k}, & EN \\ \mathbf{0}, & NEN \end{bmatrix} \\
&= \frac{\delta_k + \dfrac{\lambda_2}{N-1} \begin{bmatrix} \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} (y_k \tilde{\mathbf{x}}_k - \tilde{\mathbf{a}}^*), & EN \\ \mathbf{0}, & NEN \end{bmatrix}}{1 - \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \lambda_2 \tilde{\mathbf{I}}\right)^{-1} \tilde{\mathbf{x}}_k}.
\end{aligned}$$

Substitution of $\hat{\delta}_k^{(k)}$ into (17) with respect to notations (21) yields (19) and (20).
The theorem is proven.

The Manipulability Index in the IANC Model

Yuliya A. Veselova

Abstract Procedures aggregating individual preferences into a collective choice differ in their vulnerability to manipulations. To measure it, one may consider the share of preference profiles where manipulation is possible in the total number of profiles, which is called Nitzan–Kelly's index of manipulability. The problem of manipulability can be considered in different probability models. There are three models based on anonymity and neutrality: impartial culture model (IC), impartial anonymous culture model (IAC), and impartial anonymous and neutral culture model (IANC). In contrast to the first two models, the IANC model, which is based on anonymity and neutrality axioms, has not been widely studied. In addition, there were no attempts to derive the difference of probabilities (such as Nitzan–Kelly's index) in IC and IANC analytically. We solve this problem and show in which cases the upper bound of this difference is high enough, and in which cases it is almost zero. These results enable us to simplify the computation of indices.

Keywords Anonymity • Neutrality • IC • IANC • Manipulability

1 Introduction

A social choice rule is manipulable, if there exist at least one voter and a preference profile, such that voter can achieve a better voting result by misrepresenting his/her preferences. It is obvious that only one preference profile is enough to make a social choice rule vulnerable to manipulations. The first important result is Gibbard–Satterthwaite theorem [7, 16] that states that any non-dictatorial social choice rule with at least three possible outcomes is manipulable. Satterthwaite

Y.A. Veselova (✉)
International Laboratary of Decision Choice and Analysis, National Research University Higher School of Economics, 20 Myasnitskaya, Moscow, Russian Federation
e-mail: yul-r@mail.ru

F. Aleskerov et al. (eds.), *Clusters, Orders, and Trees: Methods and Applications: In Honor of Boris Mirkin's 70th Birthday*, Springer Optimization and Its Applications 92,
DOI 10.1007/978-1-4939-0742-7_23,

introduced the definition of a strategy-proof procedure, i.e. voting scheme in which no manipulation can occur. Since every non-dictatorial social choice rule is manipulable, the question is how to compare procedures in their vulnerability to manipulations? The first approach introduced in [13] and [9] is measuring the probability that in a randomly chosen preference profile manipulation is possible, we call this measure the Nitzan–Kelly's index. Kelly also considers an approach that takes into account the number of profiles where manipulation is very unlikely to occur, although still possible. In [10] the first method was developed and supported by computational results on the relative manipulability of social choice rules.

The authors of [2] and [1] continued this line of research. The first paper contains the results of computational experiments that reveal the degree of manipulability of social choice rules. In addition, the authors introduced some new indices for evaluating manipulability. In [1], which is fundamental to this study, manipulability is studied in two ways. First, non-singleton choice is considered, then, it extends the number of voters in the computational experiment and uses different methods of expanding preferences. All the listed articles focus on individual manipulations under impartial culture (IC) assumption. In the impartial culture model, introduced in [8], a set of all preference profiles is used for generating voters' preferences. Another important probabilistic model is the impartial anonymous culture model (IAC), first described in [11] and [6]. The question of manipulability of social choice rules in the IAC model was thoroughly investigated by [5, 12, 15], and [17]. These four publications study coalitional manipulations.

In this paper, we consider the impartial anonymous and neutral culture model (IANC), in which both names of voters and names of alternatives do not matter. In this model, some preference profiles are regarded as equivalent in terms of permutations of individuals and alternatives. Therefore, the set of all preference profiles splits up into equivalence classes. The investigation of this model was started in [3] and extended in [4]. They introduced a way of calculating the number of anonymous and neutral equivalence classes and an algorithm for their uniform random generation. However, this model has not been thoroughly analyzed yet. Particularly, a way of analyzing the difference of indices in IC and IANC without conducting a computational experiment has not been investigated in the literature. In the IC model, the Nitzan–Kelly's index is a proportion of manipulated profiles in the set of all preference profiles. In the IANC model we consider not profiles, but equivalence classes, and the Nitzan–Kelly's index in IANC is a proportion of manipulated equivalence classes.

One of the arguments for considering such model is that every unbiased social choice rule satisfies both anonymity and neutrality. It means that any two preference profiles that differ in permutation of voters and (or) names of alternatives will be both either manipulable or not with respect to those rules. We can regard an equivalence class as a type of group preference, so, considering only representatives of equivalence classes, we do not count preference profiles of the same type twice. If one looks for rules that minimize the number of public preference types, then manipulability index should be considered in IANC. The number of preference profiles grows exponentially with the number of voters and as factorial with the number

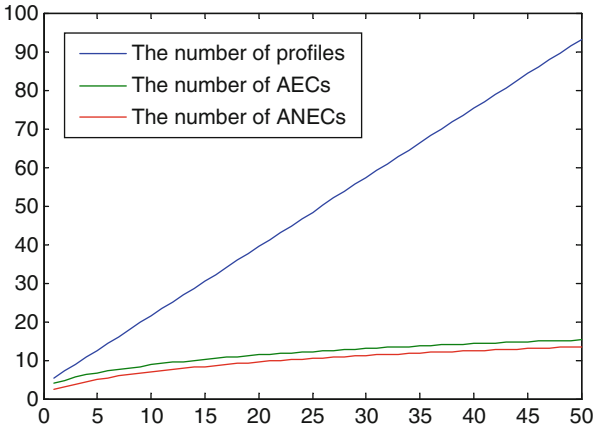


Fig. 1 The number of preference profiles, AECs and ANECs for 3 alternatives, log scale

of alternatives, while the growth of the number of anonymous equivalence classes (AECs) and anonymous and neutral equivalence classes (ANECs) is polynomial (see Fig. 1). It means that in some cases total enumeration of ANECs is possible, while the enumeration of preference profiles is not. For example, when we have 4 alternatives and 7 voters, the number of ANECs is 84,825, and the number of preference profiles is $4.586 \cdot 10^9$. In other cases, Monte-Carlo scheme in IANC will give more accurate results than in IC. To conduct computational experiments in the IANC, the algorithm for generating representatives of equivalence classes was introduced in [3]. However, we should know whether the results of computational experiments in IANC differ from those in the basic IC model. Assume that in some cases the upper bound of difference between the values of the index in IC and IANC is almost zero. Then, on the one hand, there is a plenty of results already calculated in IC, and we do not need additional computations in IANC. On the other hand, we could do computations in IANC first, because they will give more accurate results for large parameters, and put corresponding indices in IC equal to those in IANC.

Using combinatorial methods and elements of the group theory, we study properties of equivalence classes with maximal and minimal number of elements and derive the difference of indices in IANC and IC models for some cases. To illustrate it, we evaluate the maximal difference of probabilistic measures such as Nitzan–Kelly's index for the number of voters and alternatives up to 10.

We show for which number of voters and alternatives this difference is almost zero and, consequently, any probabilistic measure in the IANC model is equal to the same measure in the IC model. At the same time, for some cases this difference could be large enough to cause changes in the relative manipulability of social choice rules. We give an example of such a situation and compute the Nitzan–Kelly's indices of four social choice rules in IC and IANC for the case of three alternatives. We compare the relative manipulability of these rules and compute the difference of indices for each rule in both models. After that, we explain it in terms of the anonymous and neutral culture model.

2 The Basic Definitions and Notations

2.1 Impartial Anonymous and Neutral Culture Model

A set of alternatives consisting of m elements is denoted by A, and a set of voters is $N = \{1, 2, \ldots, n\}$ containing n elements. Preferences of the i-th individual are expressed by a linear order, P_i. A preference profile is defined as an ordered set of individual preferences $\mathbf{P} = (P_1, P_2, \ldots, P_n)$. Can also be thought as a matrix with n columns and m rows.

The set of all preference profiles with n voters and m alternatives is denoted by $\Omega(m, n)$ and has the cardinality $(m!)^n$. Impartial culture model assumes that each voter independently chooses his or her preferences out of $m!$ possible linear orders and thus, all $(m!)^n$ preference profiles are equally likely.

In the impartial anonymous culture model there is no difference between voters. Consequently, those preference profiles that differ only in the permutation of voters (or columns in the matrix representing preference profile) are regarded as the same type of collective preferences. Then we get the partition of the set $\Omega(m, n)$ into anonymous equivalence classes (AECs).

The impartial anonymous and neutral culture model assumes that both names of voters and names of alternatives do not matter. Thus, the set $\Omega(m, n)$ is divided into anonymous and neutral equivalence classes (ANECs). ANEC is a set of preference profiles that can be generated from each other by permuting voters' preferences and renaming alternatives, and every preference profile in ANEC can be taken as a root, or representative profile of this class.

The permutation of voters (or columns) is denoted by σ, which is an element of the symmetric group S_n, and a permutation on the set of alternatives is $\tau \in S_m$. These two directions in permuting preference profile are united in the pair of permutations σ and τ, which is denoted by $g = (\sigma, \tau)$. $G = S_n \times S_m$ is the group of all ordered pairs of permutations $g = (\sigma, \tau)$. G acts on the set of all preference profiles. There are $n!$ permutations of voters and $m!$ permutations of alternatives and, therefore, the number of elements in G is

$$|G| = n!m!.$$

A partition λ of n is a weakly decreasing sequence of positive integers $\lambda = (\lambda_1, \lambda_2, \lambda_3, .., \lambda_\alpha)$, such that $(\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_\alpha)$ and $\lambda_1 + \lambda_2 + \ldots + \lambda_\alpha = n$, where λ_i is a part of λ. For example, (3,1,1) and (3,2) are the partitions of 5 into 3 parts. The type of partition is denoted by $1^{\alpha_1} 2^{\alpha_2} \ldots n^{\alpha_n}$, which means that a partition λ has α_i parts of size i for each i from 1 to n. Thus, the types of (3,1,1) and (3,2) are $1^2 3^1$ and $2^1 3^1$, respectively.

Any permutation can be represented via cycle decomposition. The permutation σ defines a partition λ of n, and τ defines a partition μ of m in such a way that parts of partitions λ and μ are the lengths of cycles in σ and τ, respectively. The sum

$\alpha_1 + \alpha_2 + \ldots + \alpha_n = \alpha$ is the total number of cycles in permutation in σ. For any partition λ we define a number

$$z_\lambda = 1^{\alpha_1} 2^{\alpha_2} \ldots n^{\alpha_n} \alpha_1! \alpha_2! \ldots \alpha_n!.$$

The set of all permutations of a given cycle type $1^{\alpha_1} 2^{\alpha_2} \ldots n^{\alpha_n}$ is called a conjugacy class. The number of permutations in a conjugacy class is

$$z_\lambda^{-1} n!$$

$\mathbf{P}^g$ is the image of a profile $\mathbf{P}$ under the permutation $g = (\sigma, \tau)$. $\theta_\mathbf{P}$ is anonymous and neutral equivalence class and defined as a subset of Ω: $\{\mathbf{P}^g | g \in G\}$. Profiles $\mathbf{P}_1, \mathbf{P}_2$ are equivalent if there exists a pair of permutations $g \in G$ such that $\mathbf{P}_1^g = \mathbf{P}_2$.

$\mathbf{P}^g$ is called a fixed-point of g if for a given permutation g there exists a profile $\mathbf{P}$, such that $\mathbf{P}^g = \mathbf{P}$. A set of all fixed points for g is

$$F_g = \{\mathbf{P} \in \Omega | \mathbf{P}^g = \mathbf{P}\} \tag{1}$$

A stabilizer of $\mathbf{P}$ is a set of all permutations that do not change $\mathbf{P}$. A stabilizer of $\mathbf{P}$ is a subgroup of G and is defined as

$$G_\mathbf{P} = \{g \in G | \mathbf{P}^g = \mathbf{P}\}$$

Take any representative $\mathbf{P}$ of ANEC (θ_P). The number of elements in this equivalence class can be evaluated as a ratio

$$|\theta_\mathbf{P}| = |G|/|G_\mathbf{P}| \tag{2}$$

As usual, $GDC(\lambda)$ is the greatest common divisor of the parts of λ, $LCM(\lambda)$ is a least common multiple of the parts of λ. Binomial coefficient for an integer k, $0 \leq k \leq x$ is defined as

$$\binom{x}{k} = C_x^k = \begin{cases} \frac{x!}{k!(x-k)!}, x \in \mathbb{N} \\ x \notin \mathbb{N} \end{cases}$$

An indicator function $\chi(S)$ of statement S

$$\chi(S) = \begin{cases} 1, & \text{if } S \text{ isTrue}, \\ 0, & \text{if } S \text{ isFalse}. \end{cases}$$

The number of anonymous and neutral equivalence classes for n voters and m alternatives, $R(m, n)$, was found in [3].

$$R(m,n) = \sum_{\mu} z_{\mu}^{-1} \binom{n/t + m!/t - 1}{m!/t - 1}$$

where $t = LCM(\mu)$. This formula is simplified for n and $m!$ being relatively prime

$$R(m,n) = \frac{1}{m!} \binom{n + m! - 1}{m! - 1}$$

2.2 Manipulability

This subsection provides some definitions on manipulability. Let $\mathbf{P} = (P_1, P_2, \ldots, P_n)$ be a profile of sincere preferences. Now assume that i-th individual misrepresents his/her preferences. Such a preference profile is denoted by $\mathbf{P}_{-i} = (P_1, \ldots, P_{i-1}, P_i', P_{i+1}, \ldots P_n)$, where P_i' is the deviation of the i-th individual from his/her true preferences P_i.

Let $C(\mathbf{P})$ be the outcome of aggregating procedure on a profile $\mathbf{P}$. As in [1] we consider the case of multiple choice, which means that $C(\mathbf{P}) \subset A$. Consequently, we have to define the way of comparing subsets. For this purpose we use lexicographic method of expanding preferences, Leximin, introduced in [14]. These methods build expanded preferences on the basis of a linear order representing voter's preferences on the set of alternatives. According to the Leximin method, the worst alternatives of two sets are compared, and the set where the better alternative is contained is considered as the better set. If they are the same, then the second-worst alternatives are compared and so on. EP_i denotes the expanded preferences of individual i.

Let us take preferences xP_iyP_iz, then, according to the Leximin method,

$$\{x\}EP_i\{x,y\}EP_i\{y\}EP_i\{x,z\}EP_i\{x,y,z\}EP_i\{y,z\}EP_i\{z\}$$

Thus, manipulation is defined as follows: if for individual i $C(\mathbf{P}_{-i})EP_iC(\mathbf{P})$, then manipulation takes place.

The Nitzan–Kelly's index of manipulability is the share of manipulable profiles in the set of all preference profiles

$$NK_{\mathrm{IC}} = \frac{d_0}{(m!)^n}, \tag{3}$$

where d_0 is the number of manipulable profiles.

In the IANC we consider the ratio of the number of roots (or equivalence classes) where manipulation is possible (r_0) to the total number of roots

$$NK_{\mathrm{IANC}} = \frac{r_0}{R(m,n)}. \tag{4}$$

3 Estimating Maximal Difference of Probabilistic Measures in IC and IANC

This section provides the main theoretical results concerning impartial anonymous and neutral culture model. The aim is to estimate maximal possible difference of any probabilistic measures (such as Nitzan–Kelly's index of manipulability) in IC and IANC model. Then, if we know maximal possible difference, the actual difference will be always less. First, we consider some properties of anonymous and neutral equivalence classes. Further the problem of maximal difference is discussed in terms of manipulability, but all these results are applicable to the study of any other probability in the IC and IANC models.

First, let us reveal what properties cause this difference. Consider an abstract example of a set Ω consisting of ten preference profiles. Assume that there are four ANECs: two classes of cardinality 2, one class of cardinality 5, and the last one has only one preference profile (Fig. 2).

Then assume that only profiles from the largest equivalence class are manipulable. According to (1) and (2), the manipulability index in the IC model, NK_{IC}, is 0.5, in the IANC model $NK_{\mathrm{IANC}} = 0.25$, because only 1 of 4 equivalence classes is manipulable. The absolute difference is 0.25 and results from an inequality of equivalence classes. In the IANC model all equivalence classes are equally likely and preference profiles are not. As it can be easily seen, the manipulability index in IC exceeds (is less than) the index in IANC if the average cardinality of the set of manipulable equivalence classes exceeds (is less than) the average cardinality of the set of all equivalence classes. First, let us consider the cardinality of minimal and maximal equivalence classes. For any preference profile belonging to minimal (maximal) equivalence class, the number of stabilizing permutations is maximal (minimal). Proofs of all theorems could be found in [18] in the appendix.

Theorem 1 (Anonymous and neutral equivalence class with the minimal number of elements). *The minimal number of elements in an anonymous and neutral equivalence class is* $m!$. *This class is unique for the case of* $n \geq 3$.

Theorem 2 (Anonymous and neutral equivalence class with the maximal number of elements). *If* $m! \geq n$, *then the maximal number of elements in an anonymous and neutral equivalence class is* $m!n!$.

The number of maximal equivalence classes is not calculated precisely, but there is an estimation by the number of equivalence classes with pairwise different columns. First, we denote the set of preference profiles consisting of pairwise

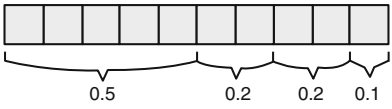


Fig. 2 A hypothetical example of four equivalence classes

different columns by $\tilde{\Omega}$. The number of equivalence classes on this set is $\tilde{R}(m,n)$. Similarly to (1), $\tilde{F}_g$ is a set of fixed points $\mathbf{P}$ from $\tilde{\Omega}$,

$$\tilde{F}_g = \{\mathbf{P} \in \tilde{\Omega} | \mathbf{P}^g = \mathbf{P}\}.$$

Lemma 1. *The number of fixed points from $\tilde{\Omega}$ for some permutation $g(\sigma,\tau)$ is equal to*

$$|\tilde{F}_g| = \begin{cases} \prod_{j=0}^{\alpha}(m! - j \cdot t), \text{ if } \lambda_1 = \lambda_2 = \ldots = \lambda_\alpha = t, \\ 0, \quad \text{otherwise.} \end{cases} \tag{5}$$

where $t = LCM(\mu)$.

The number of ANECs on $\tilde{\Omega}$ can be calculated precisely.

Theorem 3. *For any m and n such that $m! > n$, the number of equivalence classes on $\tilde{\Omega}$ is equal to*

$$\tilde{R}(m,n) = \sum_{\lambda}\sum_{\mu} z_\lambda^{-1} z_\mu^{-1} \chi(S(\lambda,\mu)) \prod_{j=0}^{\alpha-1}(m! - j \cdot t),$$

where $S(\lambda,\mu) = \{\lambda_1 = \lambda_2 = \ldots = \lambda_\alpha = t\}$.

Using Theorem 3, we estimate the number of maximal equivalence classes by the interval. This interval is very small and its bounds converge when m tends to infinity.

Corollary 1. *For any m and n such that $m! > n$ (a) The number of the maximal ANEC satisfies the following inequality*

$$\frac{2(m!-1)!}{(m!-n)!n!} - \tilde{R}(m,n) \le R_{\max}(m,n) \le \tilde{R}(m,n)$$

(b) If m and n are such that $n > m$ and n is a prime number, then the number of maximal ANEC is equal to $\tilde{R}(m,n)$.

Next we apply the results above to the problem of evaluating the maximal difference of manipulability indices. Actually, we solve it for a limited number of voters since with the growing number of voters other mechanisms work. We have already mentioned that the inequality of ANECs' cardinality causes this difference. Then, the manipulability index in IC exceeds (is less than) the index in IANC when the average cardinality of equivalence classes that are manipulable exceeds (is less than) the average cardinality of all equivalence classes. Thus, the absolute value of difference is maximal when all the classes θ, such that $|\theta| > |\theta_{\text{av}}|$, and only they are either manipulable or not manipulable. Let $max\Delta_{\text{INAC}}$ be the maximal difference of manipulability indices.

$$max\Delta_{\text{IANC}} = \left| \frac{d^*}{(m!)^n} - \frac{r^*}{R(m,n)} \right|,$$

where d^* is the number of profiles in all equivalences classes θ, such that $|\theta| > |\theta_{\text{av}}|$ (or $|\theta| < |\theta_{\text{av}}|$) and r^* is the number of such classes.

Assume that $m! > n$ and the cardinality of maximal equivalence class is $m!n!$. Here we suggest evaluating maximal difference by calculating the number of classes with cardinality that exceeds the average since for small n the only classes such that $|\theta| > |\theta_{\text{av}}|$ are the classes with a maximal number of elements.

Let n_2 be such value of n for which the second maximal cardinality of ANECs also begins to exceed the average. For example, if $m = 3$, then $n_2 = 4$; if $m = 4$, then $n_2 = 7$; if $m = 5$, then $n_2 = 14$; and if $m = 6$, $n_2 = 33$. Thus, when $n < n_2$, it is enough to know the cardinality and the number of maximal ANEC to evaluate the maximal difference of manipulability indices in IC and IANC.

In this case the difference is calculated as

$$max\Delta_{\text{IANC}} = \frac{R_{\max}(m,n) \cdot m!n!}{(m!)^n} - \frac{R_{\max}(m,n)}{R(m,n)}$$

Using Corollary 1, we get the difference in the case of $m! > n$ estimated by the interval

$$\left(\frac{2(m!-1)!}{m!-n)!n!} - \tilde{R}(m,n) \right) \cdot \left(\frac{n!}{(m!)^{n-1}} - \frac{1}{R(m,n)} \right) \leq max\Delta_{\text{IANC}}$$
$$\leq \tilde{R}(m,n) \cdot \left(\frac{n!}{(m!)^{n-1}} - \frac{1}{R(m,n)} \right)$$

In the case when m and n such that and n is a prime number an exact value of the maximal difference can be calculated strictly by

$$max\Delta_{\text{IANC}} = R_{\max}(m,n) \left(\frac{n!}{(m!)^{n-1}} - \frac{1}{R(m,n)} \right).$$

Figures 3 and 4 illustrate the difference for the number of alternatives and voters from 3 to 10. In the case when $n \geq n_2$ we calculate the number of second and third maximal equivalence classes in each case solving separate combinatorial problem. As can be seen, for the case of three and four alternatives, this difference is large enough to cause changes in the relative manipulability of social choice rules. While for the case of six or more alternatives and $n < 10$, it becomes so small and insignificant that we can take indices in the IC model equal to those in the IANC model.

It is obvious that maximal difference tends to zero, when the number of alternatives grows. However, it should be taken into account that this difference increases up to a certain value when the number of voters grows. So, we can only

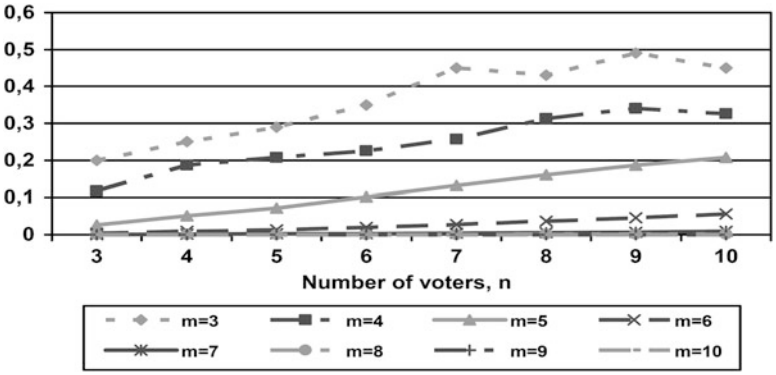


Fig. 3 Maximal difference of indices in the IC and IANC models

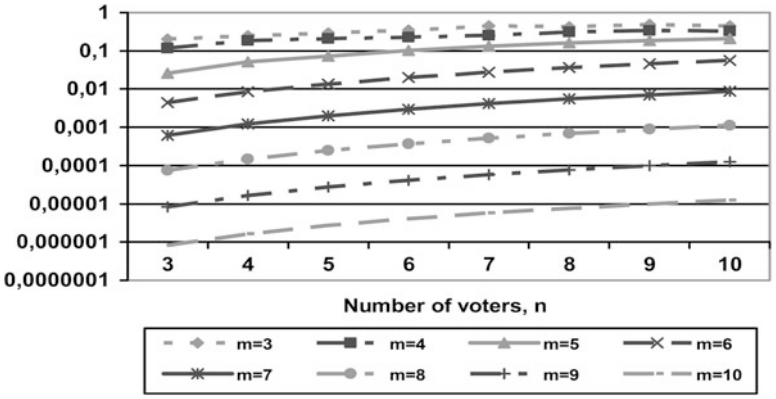


Fig. 4 Maximal difference of indices in the IC and IANC models, log scale

say that this value is not zero. If $n \ggg m!$, then equivalence classes include large numbers of profiles, and the cardinality of maximal equivalence class increases faster than the average cardinality of classes, while the minimum number of elements in equivalence class remains $m!$.

4 Calculating Manipulability of Social Choice Rules in the IANC Model

In this section we provide the results of computational experiments in IANC model of calculating manipulability indices of four social choice rules. Then we compare the difference of Nitzan–Kelly's index in IC and IANC with maximal difference

estimated in the previous section. The rules considered are plurality rule, approval voting (with fixed quota of 2), Borda's rule, and Black's procedure.

1. *Plurality Rule.* This rule chooses the alternative which is the best for the maximal number of voters.

$$a \in C(\mathbf{P}) \Leftrightarrow [\forall x \in A, n^+(a, \mathbf{P}) \geq n^+(x, \mathbf{P})]$$

   where $n^+(x, \mathbf{P}) = card\{i \in N \mid \forall y \in A, xP_i y\}$.
2. *Approval Voting.* Social choice is an alternative at the place of q or higher in the preferences of the maximal number of voters.

$$a \in C(\mathbf{P}) \Leftrightarrow [\forall x \in A, n^+(a, \mathbf{P}, q) \geq n^+(x, \mathbf{P}, q)]$$

   where $n^+(x, \mathbf{P}, q) = card\{i \in N \mid card\{y \in A \mid xP_i y\} \geq q - 1\}$.
3. *Borda's Rule.* For each alternative in the i-th individual preferences the number $r_i(x, \mathbf{P})$ is calculated as follows:

$$r_i(x, \mathbf{P}) = card\{b \in A \mid xP_i b\}.$$

   The sum of $r_i(x, \mathbf{P})$ over all $i \in N$ is called a Borda's count.

$$r(x, \mathbf{P}) = \sum_{i=1}^{n} r_i(x, \mathbf{P})$$

   Borda's rule chooses an alternative with the maximal Borda's count.

$$a \in C(\mathbf{P}) \Leftrightarrow [\forall x \in A, r(a, \mathbf{P}) \geq r(x, \mathbf{P})]$$

4. *Black's Procedure.* Chooses a Condorset winner, if it exists, and, if it does not exist, the winner of Borda's rule is chosen.

We restrict our scope to the case of three alternatives. First, we compute the Nitzan–Kelly's indices in the impartial culture model (Fig. 5), impartial anonymous and neutral culture model (Fig. 6) using the Leximin method. After that, we calculate the difference of these indices

$$\Delta NK_{\text{IANC}} = \frac{d_0}{(m!)^n} - \frac{r_0}{R(m, n)},$$

which is represented in Fig. 7.

The maximal difference is lowest and the highest boundary on Fig. 7. The difference of manipulability indices is negative only for approval voting rule. This fact can be explained as follows: preference profiles in which manipulation is possible often belong to equivalence classes with a small number of elements.

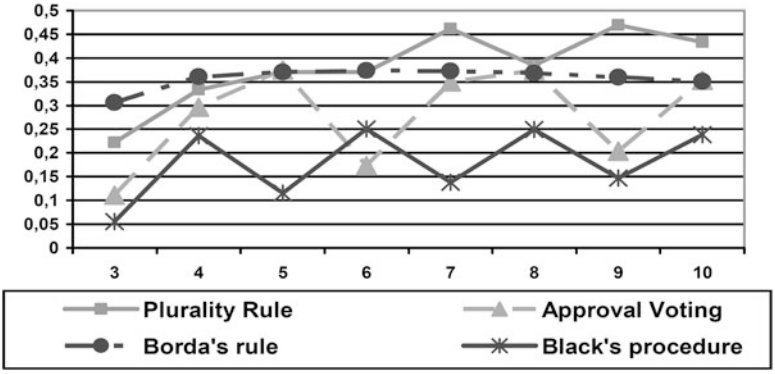


Fig. 5 The Nitzan–Kelly's index for the Leximin method in the IC model

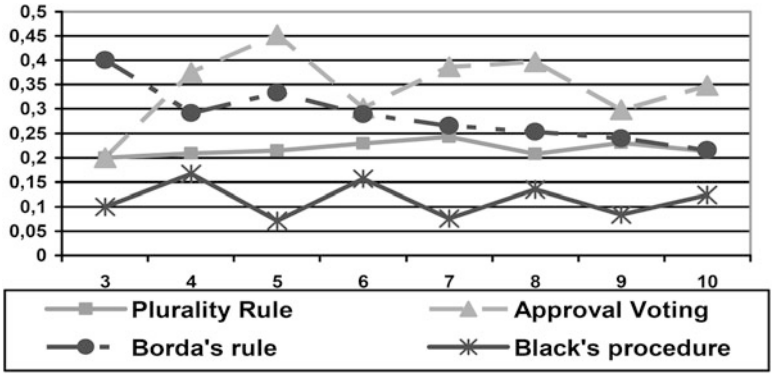


Fig. 6 The Nitzan–Kelly's index for the Leximin method in the IANC model

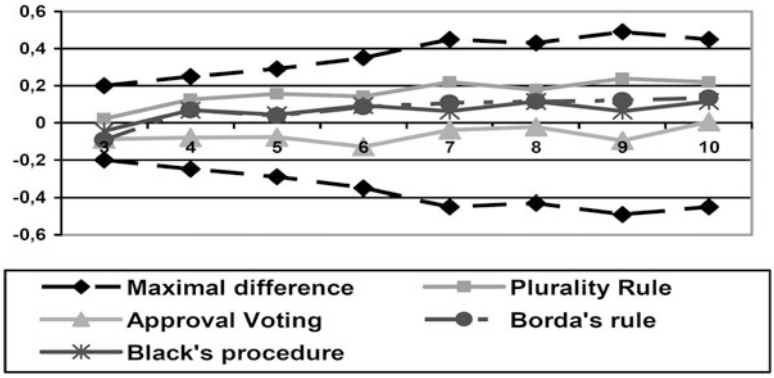


Fig. 7 The difference of the Nitzan–Kelly's index in IC and IANC, Leximin

The plurality rule has the highest level of difference for $3 \leq n \leq 10$. These two facts cause the changes in the relative manipulability of social choice rules. The approval rule turns out to be the most manipulable in the IANC model, while under the IC assumption it is the second least manipulable rule. The relative manipulability of the plurality rule and approval voting rule changed to the opposite in most cases. However, Black's procedure is the least manipulable in both cultures.

5 Concluding Remarks

Anonymity and neutrality are the basic axioms in social choice theory. The IANC model, based on these axioms, assumes that both names of voters and names of alternatives do not matter. In the IC model, the Nitzan–Kelly's index is the probability that any preference profile independently drawn from the set of all preference profiles will be manipulable. In the IANC model, it is the same probability on the set of anonymous and neutral equivalence classes. The representatives of equivalence classes could be considered as "types" of public preferences. Consequently, minimizing manipulability index in IANC means minimizing types of preferences that allow manipulations.

We study to what extent the value of manipulability index in IANC could differ from index in IC. We reveal some properties of IANC model. Using methods of combinatorics and group theory, we evaluate the number and cardinality of anonymous and neutral equivalence classes with a maximal and minimal number of elements. Then we estimate the maximal possible difference between Nitzan–Kelly's index (and, consequently, any other probabilistic measure) for small number of voters and conclude that it is not zero for large number of voters. At the same time, maximal difference tends to zero, when the number of alternatives grows and increases up to a certain positive value with growing number of voters.

This theoretical study allows us to avoid additional highly complex computations, when indices in IC model are equal to the same indices in IANC. To illustrate such cases when transition from IC to IANC can change the situation, we analyze the actual difference of manipulability indices of four social choice rules in the IC and IANC models with Leximin extension method.

Acknowledgments The author is grateful to F. Aleskerov for his scientific advice and encouragement and also would like to thank D. Piontkovsky and D. Shvarts for their helpful comments. Support from the Basic Research Program of the National Research University Higher School of Economics is gratefully acknowledged.

References

1. Aleskerov, F., Karabekyan, D., Sanver, M.R., Yakuba, V.: On the degree of manipulability of multi-valued social choice rules. Homo Oeconomicus **28**(1/2), 205–216 (2011)
2. Aleskerov, F., Kurbanov, E.: Degree of manipulability of social choice procedures. In: Alkan, et al. (eds.) Current Trends in Economics, P. 13–28. Springer, Berlin/Heidelberg/New York (1999)
3. Egecioglu, O.: Uniform Generation of Anonymous and Neutral Preference Profiles for Social Choice Rules. Technical Report TR2005-25, Department of Computer Science, UCSB (2005)
4. Egecioglu, O., Giritligil, A.E.: Public preference structures with impartial anonymous and neutral culture model. Monte Carlo Method Appl. **15**(3), 241–255 (2009)
5. Favardin, P., Lepelley, D.: Some further results on the manipulability of social choice rules. Soc. Choice Welfare **26**, 485–509 (2006)
6. Gehrlein, W.V., Fishburn, P.C.: Condorcet's paradox and anonymous preference profiles. Publ. Choice **26**, 1–18 (1976)
7. Gibbard, A.: Manipulation of voting schemes. Econometrica **41**, 587–601 (1973)
8. Guilbaud, G.T.: Les theories de l'interet general et le problemelogique de l'agregation. Econ. Appl. **5**, 501–584 (1952)
9. Kelly, J.: Minimal manipulability and local strategy-proofness. Soc. Choice Welfare **5**, 81–85 (1988)
10. Kelly, J.: Almost all social choice rules are highly manipulable, but few aren't. Soc. Choice Welfare **10**, 161–175 (1993)
11. Kuga, K., Nagatani, H.: Voter antagonism and the paradox of voting. Econometrica **42**(6), 1045–1067 (1974)
12. Lepelley, D., Valognes, F.: Voting rules, manipulability and social homogeneity. Publ. Choice **116**(1/2), 165–184 (2003)
13. Nitzan, S.: The vulnerability of point-voting schemes to preference variation and strategic manipulation. Publ. Choice **47**, 349–370 (1985)
14. Pattanaik, P.: Strategy and Group choice. North-Holland, Amsterdam (1978)
15. Pritchard, G., Wilson, M.: Exact results on manipulability of positional voting rules. Soc. Choice Welfare **29**, 487–513 (2007)
16. Satterthwaite, M.: Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. J. Econ. Theory **10**, 187–217 (1975)
17. Slinko, A.: How the size of a coalition affects its chances to influence an election. Soc. Choice Welfare **26**, 143–153 (2006)
18. Veselova, Y.: The difference between manipulability indices in IC and IANC models. EC 'Economics', Higher School of Economics, preprint (2012)